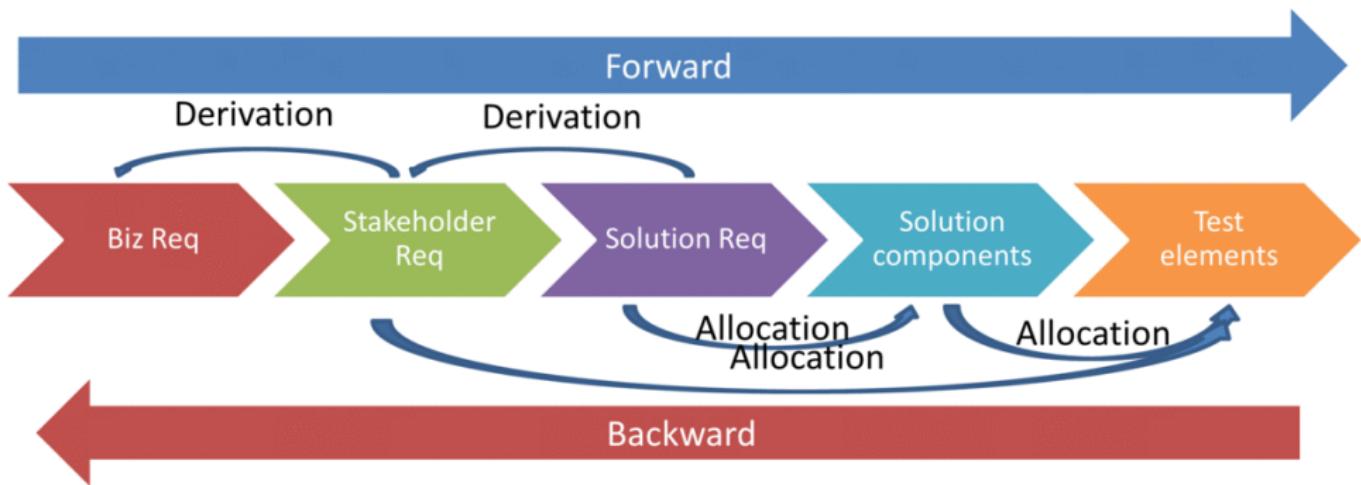


近期发表的利用LLM开展需求追踪/追溯的研究

Rachel

公众号：学海拾珠漫步知途

2025-06



- ① 文献检索结果
- ② Improving Software Development Traceability with Structured Prompting
- ③ LiSSA: Toward Generic Traceability Link Recovery through Retrieval-Augmented Generation
- ④ Code Gradients: Towards Automated Traceability of LLM-Generated Code
- ⑤ TVR: Automotive System Requirement Traceability Validation and Recovery Through Retrieval-Augmented Generation
- ⑥ On the Impact of Requirements Smells in Prompts: The Case of Automated Traceability
- ⑦ Requirements Traceability Link Recovery via Retrieval-Augmented Generation
- ⑧ Establishing Traceability Between Natural Language Requirements and Software Artifacts by Combining RAG and LLMs
- ⑨ LLM-based Approach to Automatically Establish Traceability between Requirements and MBSE
- ⑩ An LLM-based Approach to Recover Traceability Links between Security Requirements and Goal Models
- ⑪ Classification or Prompting: A Case Study on Legal Requirements Traceability

检索过程

- 检索条件：使用关键词“*LLM traceability*”在Google Scholar中检索文献，浏览前5页结果；
- 检索范围：CCF C类及以上级别的会议或者期刊，JCR Q2以上级别的期刊；
- 发表时间：2024年1月至2025年5月；

检索结果及总体定性分析

共检索到10篇论文，包括一篇是CCF A类会议ICSE主旨论文，一篇ICSE 伴会论文，两篇Lionel组的working paper。

总体趋势：不温不火，非热门话题，属于需求工程领域的一个子话题。

任务特点：

① source artifacts和target artifacts之间是异构(例如，自然语言需求文档和建模语言的设计模型)、异质(例如，自然语言的需求文档和编程语言的源代码、测试工件)。artifacts的这种多源、异构、异质的特性为该任务带来如下几个“卡脖子”挑战：[语义鸿沟 \(semantic gap\)](#)，自然语言之间的鸿沟，例如，系统需求(system requirements)和用户需求(user/stakeholder requirements/needs)之间等不同level间需求必然存在的语义鸿沟；

② 需要project-level上下文，没有对整个项目的了解和理解，只看一对儿工件，是很难准确判断这两者之间是否应该建立追踪链接，就好比NLP中，理解一个词语的语义需要依赖上下文，此任务中也是需要[项目背景](#)这一上下文；

③ 组合爆炸，需要判别 $M \times N$ 个(source artifact, target artifact) pairs之间的是否应该建立链接，这就需要一些[预过滤](#)技术，此外，可能还需要建立双向链接，即，backward和forward，需要判别的数量进一步增加。

检索结果列表—10篇 I

- [1] Bonner, Maria, Gabor Schulz, Marc Zeller, and Ana Savu. "LLM-based Approach to Automatically Establish Traceability between Requirements and MBSE." *INCOSE International Symposium* 34, no. 1 (2024): e13285. DOI: 10.1002/iis.2.13285.
- [2] Hassine, Jameleddine. "An LLM-based Approach to Recover Traceability Links between Security Requirements and Goal Models." In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024)*, 643-651. New York, NY: ACM, 2024. DOI: 10.1145/3661167.3661261.
- [3] North, Marc, Amir Atapour-Abarghouei, and Nelly Bencomo. "Code Gradients: Towards Automated Traceability of LLM-Generated Code." In *Proceedings of the 2024 IEEE 32nd International Requirements Engineering Conference (RE)*, 321-328. Piscataway, NJ: IEEE, 2024. DOI: 10.1109/RE59067.2024.00038.
- [4] Ali, Syed Juned, Varun Naganathan, and Dominik Bork. "Establishing Traceability Between Natural Language Requirements and Software Artifacts by Combining RAG and LLMs." In *Conceptual Modeling. ER 2024*, edited by W. Maass et al., 295-314. Lecture Notes in Computer Science, vol. 15238. Cham: Springer Nature Switzerland AG, 2025. DOI: 10.1007/978-3-031-75872-0_16.

检索结果列表—10篇 II

- [5] Etezadi, Romina, Chetan Arora, Sallam Abualhaija, and Lionel Briand. “Classification or Prompting: A Case Study on Legal Requirements Traceability.” arXiv preprint arXiv:2502.0 (2025).
- [6] Fuchß, Dominik, Tobias Hey, Jan Keim, Haoyu Liu, Niklas Ewald, Tobias Thirolf, and Anne Koziolek. “LiSSA: Toward Generic Traceability Link Recovery through Retrieval-Augmented Generation.” In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering (ICSE '25)*. Forthcoming. New York, NY: ACM / Piscataway, NJ: IEEE, 2025.
- [7] Hey, Tobias, Dominik Fuchß, Jan Keim, and Anne Koziolek. “Requirements Traceability Link Recovery via Retrieval-Augmented Generation.” In *Requirements Engineering: Foundations for Software Quality. REFSQ 2025*, edited by Shazia Sadiq, Wil M.P. Van der Aalst, Yushun Xia, and Liang Julius Zhang, 1-16. Lecture Notes in Computer Science, vol. 15012. Cham: Springer, 2025.
- [8] Kim, Dae-Kyoo. “Improving Software Development Traceability with Structured Prompting.” *Journal of Computer Information Systems* (2025). DOI: 10.1080/08874417.2025.
- [9] Niu, Feifei, Hanyang Hu, Rongqi Pan, Lionel C. Briand, and Krishna Koravadi. “TVR: Automotive System Requirement Traceability Validation and Recovery Through Retrieval-Augmented Generation.” arXiv preprint arXiv:2504.15427v1 (2025).

检索结果列表—10篇 III

- [10] Vogelsang, Andreas, Alexander Korn, Giovanna Broccia, Alessio Ferrari, Jannik Fischba and Chetan Arora. “On the Impact of Requirements Smells in Prompts: The Case of Automated Traceability.” arXiv preprint arXiv:2501.04810v1 (2025).

1. 标题

通过结构化提示提升软件开发可追溯性

2. 作者及单位

Dae-Kyoo Kim

- 奥克兰大学，罗切斯特，密歇根州，美国

3. 文献来源

Kim, Dae-Kyoo. “*Improving Software Development Traceability with Structured Prompting.*” Journal of Computer Information Systems (2025): 1-21.

4. 文献类型与关键词

文章类型：实证研究、方法学研究

英文关键词：AI models; LLM; Prompt; software development; Traceability

中文关键词：人工智能模型；大语言模型；提示；软件开发；可追溯性

研究动机: 生成式大型语言模型（LLM）已广泛嵌入软件开发生命周期，但由其生成的工件常出现**有效性不足、前后不一致与信息缺失**等问题，导致开发阶段间的可追踪性受损。现有研究很少系统探讨“**如何通过设计高质量提示（prompt）来提升软件开发可追踪性**”，亟需解决这一空白，以减少人工校验成本并提高软件质量。

研究目标: 提出一套面向软件开发场景的结构化提示设计方案，并验证其在提升可追踪性方面的有效性。

- 萃取并改造社会科学领域的问卷/访谈设计原则，形成软件工程提示设计原则；
- 构建包含“Prompt ID、Context、Input Data、Prompt、Expected Outcome”等栏位的结构化提示模板；
- 定义活动级（TMA）、阶段级（TMP）、过程级（TMPr）三层可追踪性度量指标体系；
- 通过案例研究评估结构化提示对不同LLM与开发阶段的增益效果。

研究问题:

- 在软件开发中，什么样的原则应指导结构化提示的设计？
- 应如何设计提示的结构以增强软件开发的可追溯性？
- 结构化提示在改善软件开发可追溯性方面的有效性如何？

研究方法

本研究采用了一种结合了理论构建、度量体系设计和实证评估的综合研究方法。该方法具体包含以下四个步骤：

- **原则与设计构建:** 研究首先从社会科学领域的访谈和问卷设计理论中汲取灵感，改编并提出了一套适用于软件开发提示设计的核心原则，包括清晰性、客观性、特异性等。基于这些原则，研究设计了一个包含五个部分的结构化提示模板：提示ID(PromptID)、上下文(Context)、输入数据(InputData)、提示(Prompt)和预期输出(ExpectedOutcomes)。
- **可追溯性度量定义:** 为了量化评估效果，研究建立了一套分层的可追溯性度量体系。该体系从有效性(validity)、完整性(completeness)和一致性(consistency)三个维度来定义可追溯性，并提出了三个具体的度量指标：活动级可追溯性度量(TMA)、阶段级可追溯性度量(TMP)和过程级可追溯性度量(TMPR)。
- **案例研究与实验执行:** 研究通过一个智能家居系统开发的案例研究来评估所提出方法的有效性。实验覆盖了需求分析、设计建模、实现和测试四个阶段中的十项具体开发活动。研究选用了Meta、Gemini、Copilot、Claude、ChatGPT五种主流AI模型，并让它们分别使用结构化提示和非结构化提示来完成所有开发活动。
- **数据分析与比较:** 研究使用预定义的度量指标 (TMA, TMP, TMPR) 对两种提示方法生成的工件进行量化评估。最后，通过对比分析两种方法在不同开发阶段和不同AI模型上的表现，系统性地评估结构化提示的有效性。

研究结果

本研究的评估结果清晰地表明了结构化提示的优越性，主要研究结果如下：

- **总体有效性：**与非结构化提示相比，结构化提示在所有五个AI模型上均显著提升了软件开发过程的整体可追溯性，过程级可追溯性度量（TMPR）的平均改进幅度达到了23.92%。
- **分阶段效果：**结构化提示在设计建模(DesignModeling)阶段的效果最为显著，平均改进了32.21%。其次是测试(Testing)阶段(26.35%)、实现(Implementation)阶段(21.38%)需求分析(RequirementsAnalysis)阶段(19.14%)。
- **分模型效果：**所有参与测试的AI模型都从结构化提示中受益。其中，**Gemini的改进幅度最大，达到了47.42%**；而Meta的改进幅度最小，为14.24%。
- **输出稳定性：**结构化提示不仅提升了可追溯性，还带来了更稳定和可预测的输出。其TMPR得分的标准差(0.045)远低于非结构化提示(0.084)。
- **识别的挑战：**研究也发现，即使使用了详细的结构化提示，AI模型有时仍未能完全遵守指令，导致输出不完整。此外，部分模型（如Claude）表现出过度规范化的倾向，即利用其内部领域知识输出了超出需求范围的内容。

研究结论

结构化提示设计在多种LLM和开发阶段均显著提升软件工件可追踪性，验证了“高质量提示 = 高质量工件”的核心命题；同时提示原则与度量体系为后续LLM-辅助软件工程研究提供了可复用框架。作者亦指出须进一步研究提示约束机制与跨领域、复杂项目的泛化能力，以充分释放LLM在软件工程中的潜力。

启发：① 结构化提示可以提升整体可追溯性，说明我们的提示方法事有一定的依据的，我们提示方法生成的数据是有一定质量保障的，即，结构化提示不仅提升了可追溯性，还带来了更稳定和可预测的输出；② 为什么用Gemini做数据生成，而不用其他LLM呢，这篇论文的实验结果为我们提供了依据；

1. 标题

LiSSA: 面向通用可追踪链接恢复的检索增强生成方法

2. 作者及单位

Dominik Fuchß; Tobias Hey; Jan Keim; Haoyu Liu; Niklas Ewald; Tobias Thirolf; Anne Koziolek

- 德国卡尔斯鲁厄理工学院（Karlsruhe Institute of Technology, KIT）CASTEL 信息安全与可靠性研究所

3. 文献来源

Fuchß, Dominik, et al. "*LiSSA: Toward Generic Traceability Link Recovery through Retrieval-Augmented Generation.*" Proceedings of the IEEE/ACM 47th International Conference on Software Engineering. ICSE. Vol. 25. 2025.

4. 文献类型与关键词

文章类型：方法学研究+实证研究

英文关键词：Traceability Link Recovery; Large Language Models;
Retrieval-Augmented Generation

中文关键词：可追踪性链接恢复；大型语言模型；检索增强生成

研究动机

本研究的动机源于对当前软件工程实践中一个核心挑战的深刻洞察：现有可追溯性链接恢复（Traceability Link Recovery, TLR）方法的局限性与新兴大语言模型（LLM）技术带来的机遇之间的矛盾。

- **现有方法的“专”与“弱”：**在软件开发过程中，理解需求、文档、模型、代码等多种工件间的复杂关联至关重要。自动化的TLR技术是揭示这些关联的关键，能支持变更影响分析、缺陷定位等重要任务。然而，当前主流的自动化方法通常是“任务专用”的，即为特定类型的工件对（如需求到代码）设计，缺乏通用性。更重要的是，这些方法的性能表现不一，且常常未达到工业实践所要求的精度和效率。
- **LLM的潜力与挑战：**LLM在自然语言理解方面展现出巨大潜力，已在代码生成、代码摘要等多种软件工程任务中证明了其有效性。这预示着LLM有能力成为实现一种通用TLR方法的理想技术基础。然而，一个核心挑战在于，LLM本身不具备特定项目的上下文知识，并且由于输入长度的限制，无法直接处理大型软件项目的全部内容。如何高效地为LLM提供执行TLR任务所需的精确信息，是一个亟待解决的非平凡问题。
- **RAG作为可行路径：**检索增强生成（Retrieval-Augmented Generation, RAG）技术为此挑战提供了一条可行的解决路径。该技术首先通过信息检索（IR）技术找到相关的工件片段，然后将这些检索到的内容提供给LLM，辅助其生成最终判断。

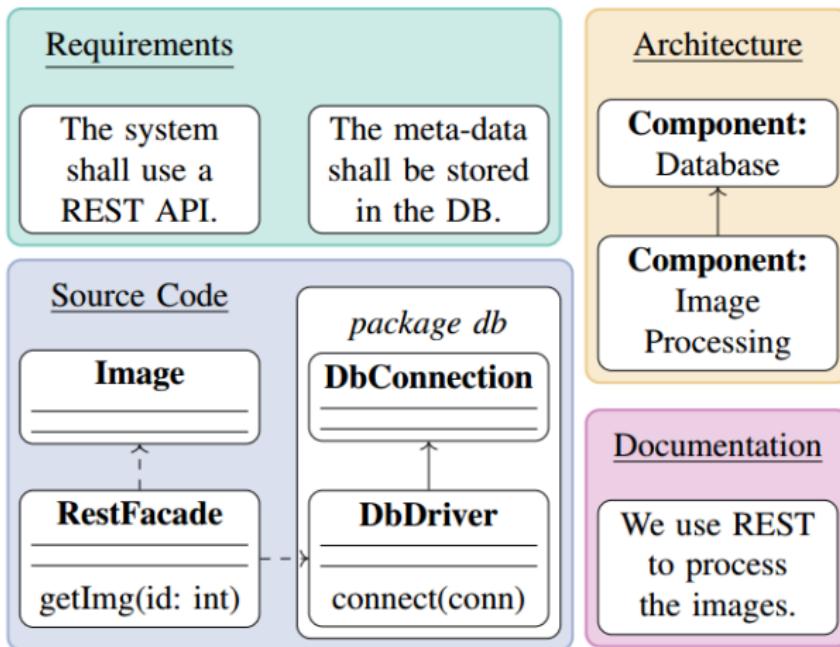


Fig. 1. Example Project with Requirements, an Architecture Diagram, Natural Language Documentation, and Source Code

研究目标

基于上述动机，本研究的核心目标是提出并验证一个基于RAG的通用TLR框架，旨在克服现有专用方法的局限性，并探索LLM在多类型软件工件间进行可追溯性链接恢复的有效性。为实现这一目标，研究工作具体聚焦于：

- 设计一个通用框架(LiSSA):构建一个名为LiSSA的框架,该框架能够集成RAG技术,利用LLM的语言理解能力,对不同类型的软件工件(如需求、代码、架构文档、架构模型)执行TLR任务。
- 进行广泛的实证评估:在三个覆盖不同场景的TLR任务上对LiSSA进行系统的实证评估,这些任务分别是:需求到代码、架构文档到代码、以及架构文档到架构模型。
- 与先进技术进行对比:将LiSSA的性能与各个任务领域内最先进的(state-of-the-art)专用TLR方法进行严格比较,以客观评价其相对优势与不足。

研究问题

为系统性地达成研究目标，本研究设定了以下四个具体的研究问题
(Research Questions, RQs):

- RQ1: 一个通用的、基于RAG的TLR方法是否优于任务专用的、最先进的方法？
- RQ2: 在TLR任务中，思维链（Chain-of-thought, CoT）提示技术是否比简单的分类提示更有效？
- RQ3: 与在整个工件级别进行映射相比，在细粒度的子工件级别上进行检索和映射，能否提升TLR的性能？
- RQ4: 与仅基于嵌入的信息检索（IR）方法相比，RAG（即增加LLM分类步骤）能否提升TLR的性能？

研究方法

本研究采用了一种结合框架构建与实证评估的研究方法，其核心是LiSSA框架的设计与验证。

LiSSA框架设计：该框架遵循一个三阶段流程。

- 阶段一：预处理与嵌入 (Preprocessing & Embedding): LiSSA首先将输入的各类工件（自然语言文档、代码、模型）通过特定的预处理器分解为更小、更易于处理的“元素”(Elements)，例如将文档分解为句子，将代码分解为方法。这些元素可以是不同粒度的，如文件级或方法级。随后，使用嵌入模型（text-embedding-3-large）将每个元素的文本表示转换为高维向量，并存入向量数据库。

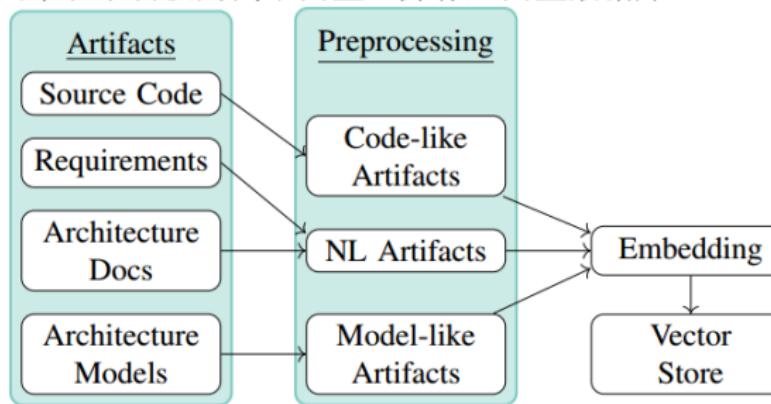


Fig. 3. Preprocessing & Embedding of Artifacts (NL for Natural Language)

- 阶段二: 检索 (Retrieval): 对于每一个源工件的元素, 该阶段利用向量间的余弦相似度, 从目标工件的元素集合中检索出最相似的Top-k个候选元素。这一步有效地将搜索空间缩小到最相关的候选范围内。
- 阶段三: 映射 (Mapping): 此阶段包含两个子步骤。首先是提示 (Prompting), LiSSA将源元素与每个候选目标元素组合成一个提示, 提交给LLM (GPT-4o或GPT-4o mini) 进行分类, 判断两者是否存在关联。研究中测试了两种提示策略: 一种是简单的零样本分类提示 (KISS), 另一种是引导LLM进行推理的思维链 (CoT) 提示。其次是聚合 (Aggregation), 将在元素级别上被LLM判断为相关的链接, 根据任务要求的粒度 (例如, 从句子-方法级别聚合到文件-文件级别) 进行整合, 最终生成追溯链接。

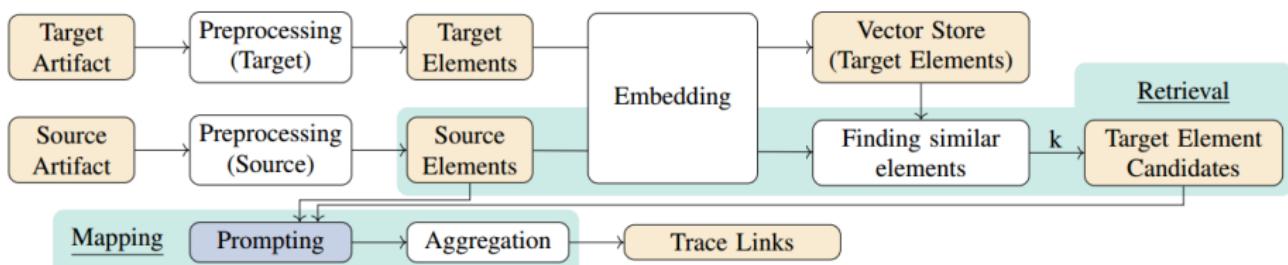


Fig. 5. Overview of the Retrieval & Mapping. Data is in orange, prompting is in blue, and other processing is in white.

实证评估设计

- 任务与数据集：研究选取了三个TLR任务，并全部采用文献中公开的、已建立基准的数据集，以减少偏见并确保可比性。
- 性能度量：使用TLR领域公认的Precision（精确率）、Recall（召回率）以及 F_β -score（重点评估F1和F2分数）作为核心评价指标。
- 基线对比：将LiSSA与每个任务上最先进的专用方法（如FTLR、COMET、ArDoCode）以及一个仅包含检索步骤的“Retrieval-Only”基线进行比较。
- 统计检验：采用双边Wilcoxon符号秩检验来评估最佳配置结果与基线之间的差异是否具有统计显著性（ $\alpha=0.05$ ）。

TABLE I
OVERVIEW OF DATASET FOR REQUIREMENTS TO CODE TLR. IT MARKS
ITALIAN AND ENGLISH NATURAL LANGUAGE DESCRIPTIONS, AND
TL STANDS FOR TRACE LINK

Dataset	Domain	Language		# of Artifacts		
		NL	Programming	Req	Code	TL
SMOS	Education	IT	Java	67	100	1044
eTour	Tourism	EN	Java	58	116	308
iTrust	Healthcare	EN	Java	131	226	286
Dronology (RE)	Aerospace	EN	Java, Python	99	423	602
Dronology (DD)	Aerospace	EN	Java, Python	211	423	740

TABLE II
NUMBER OF ARTIFACTS PER TYPE AND NUMBER OF TRACE LINKS IN
THE GOLD STANDARD FOR EACH ARCHITECTURE TRACEABILITY
PROJECT. AD STANDS FOR ARCHITECTURE DOCUMENTATION.

Artifact Type	MS	TS	TM	BBB	JR
AD Sentences	37	43	198	85	13
Architecture Model Elements	23	19	16	24	6
Source Code Files	97	205	832	547	1,979
AD to Model Trace Links	29	27	51	52	18
AD to Code Trace Links	50	707	7,610	1,295	8,240

Prompt 1: KISS

Question: Here are two parts of software development artifacts.

{source_type}: "{source_content}"

{target_type}: "{target_content}"

Are they related?

Answer with 'yes' or 'no'.

Prompt 2: Chain-of-thought

Below are two artifacts from the same software system. Is there a traceability link between (1) and (2)? Give your reasoning and then answer with 'yes' or 'no' enclosed in <trace> </trace>.

(1) {source_type}: "{source_content}"

(2) {target_type}: "{target_content}"

研究结果

对于**RQ1 (RAG vs. SOTA)**: 研究结果表明，基于RAG的LiSSA方法在代码相关的TLR任务上表现出色。具体而言，它在需求到代码的TLR任务上显著优于所有最先进的基线方法。

TABLE IV
COMPARISON OF THE AVERAGE AND WEIGHTED AVERAGE F_β -SCORES
USING GPT-4o AND GPT-4o MINI FOR REQUIREMENT-TO-CODE TLR
USING TOP-20 RETRIEVAL

Approach	Avg.				w. Avg.					
	Prep.	Cls.	GPT-4o		GPT-4o mini		GPT-4o		GPT-4o mini	
			F ₁	F ₂						
None / None	No		.230	.332	.230	.332	.249	.332	.249	.332
	P1		.312	.369	.247	.345	.288	.319	.260	.334
	P2		.322	.368	.295	.356	.299	.319	.282	.315
Sentence / Chunk(200)	No		.203	.305	.203	.305	.235	.332	.235	.332
	P1		.242	.296	.210	.309	.256	.288	.241	.331
	P2		.242	.290	.236	.317	.257	.279	.261	.325
Sentence / +Art.	+Art.		.261	.327	.243	.318	.277	.320	.263	.314
	No		.198	.283	.198	.283	.243	.321	.243	.321
	P1		.257	.294	.211	.292	.285	.299	.256	.325
Method / +Art.	P2		.257	.297	.232	.295	.289	.305	.268	.307
	+Art.		.260	.309	.238	.296	.288	.312	.267	.298
Baselines			F ₁		F ₂		F ₁		F ₂	
VSM _{OPT}			.282		.256		.283		.257	
LSI _{OPT}			.285		.270		.285		.268	
FTLR			.278		.300		.273		.277	
FTLR _{OPT}			.303		.327		.294		.329	

在文档到代码的任务中，对于小型项目，LiSSA的F1分数也显著优于基线；但对于大型项目，其性能则有所下降。

TABLE VI
COMPARISON OF THE AVERAGE AND WEIGHTED AVERAGE F_β -SCORES
FOR SOFTWARE ARCHITECTURE DOCUMENTATION TO CODE TLR ON
THE SMALLER PROJECTS BIGBLUEBUTTON, TEASTORE, MEDIASTORE
USING GPT-4O MINI AND TOP-20 RETRIEVAL

Preprocessor	Classifier	Approach		Avg.		w. Avg.	
		F ₁	F ₂	F ₁	F ₂	F ₁	F ₂
Sentence / None	No LLM	.209	.257	.240	.260		
	Prompt 1	.212	.259	.243	.262		
	Prompt 2	.217	.255	.249	.253		
Sentence / Chunk(200)	No LLM	.206	.238	.214	.204		
	Prompt 1	.206	.238	.214	.204		
	Prompt 2	.213	.241	.221	.203		
Sentence / Chunk(1000)	No LLM	.196	.230	.213	.206		
	Prompt 1	.196	.228	.213	.205		
	Prompt 2	.205	.237	.217	.203		
Baselines		F ₁	F ₂	F ₁	F ₂		
ArDoCode		.178	.302	.189	.318		

然而，在文档到架构模型的TLR任务中，LiSSA的性能未能超越为该任务量身定制的专用方法ArDoCo。

TABLE VII
COMPARISON OF THE AVERAGE AND WEIGHTED AVERAGE F_β -SCORES
FOR ARCHITECTURE MODEL TO ARCHITECTURE DOCUMENTATION
(M2D) AND ARCHITECTURE DOCUMENTATION TO ARCHITECTURE
MODEL (D2M) TLR USING TOP-10 RETRIEVAL AND GPT-4O MINI

Features	Approach	Avg.				w. Avg.				
		D2M		M2D		D2M		M2D		
		Cl.	F ₁	F ₂	F ₁	F ₂	F ₁	F ₂	F ₁	
Name	No		.162	.305	.350	.505	.131	.259	.334	.466
	P1		.173	.323	.357	.512	.140	.275	.340	.472
	P2		.286	.467	.458	.589	.234	.404	.424	.534
Name, Interfaces	No		.162	.305	.349	.504	.131	.259	.334	.466
	P1		.169	.317	.355	.509	.138	.271	.339	.471
	P2		.274	.447	.450	.586	.223	.383	.419	.534
Name, Interfaces, Usages	No		.161	.303	.341	.494	.130	.257	.322	.452
	P1		.169	.315	.348	.501	.137	.269	.329	.458
	P2		.265	.435	.417	.554	.215	.373	.384	.499
Baselines			F ₁		F ₂		F ₁		F ₂	
ArDoCo			.822		.814		.802		.806	

对于RQ2 (CoT vs. 简单提示): 评估发现，在F1分数上，采用CoT提示的配置在平均表现上优于使用简单分类提示 (KISS prompt) 的配置。这说明引导LLM进行逻辑推理有助于提升链接判断的准确性。

对于RQ3 (细粒度 vs. 工件级): 对于需求到代码的TLR任务，将工件分解为细粒度元素 (如句子或代码块) 进行检索和映射，在平均性能上并没有带来提升。最佳的预处理策略是项目相关的，不存在一个普遍适用的最优配置。

TABLE III

RESULTS FOR REQUIREMENT-TO-CODE TLR WITH GPT-4O & TOP-20 (PREP. FOR PREPROCESSOR, CLS. FOR CLASSIFIER, ART. FOR ARTIFACTS)

Approach	SMOS				eTour				iTrust				Dronology (RE)				Dronology (DD)				
	P	R	F ₁	F ₂	P	R	F ₁	F ₂	P	R	F ₁	F ₂	P	R	F ₁	F ₂	P	R	F ₁	F ₂	
VSM _{OPT}	.430	.414	.422	.417	.557	.427	.483	.448	.208	.227	.217	.223	.844	.087	.158	.106	.846	.071	.131	.087	
LSI _{OPT}	.415	.430	.422	.427	.452	.453	.453	.453	.251	.255	.253	.254	.333	.107	.162	.124	.757	.074	.135	.090	
COMET _{OPT}	.195	.572	.291	.413	.410	.468	.437	.455	.361	.231	.282	.249	—	—	—	—	—	—	—	—	
FTLR	.444	.331	.380	.349	.379	.633	.474	.558	.165	.339	.222	.280	.183	.161	.172	.165	.129	.154	.140	.148	
FTLR _{OPT}	.314	.588	.409	.501	.505	.597	.548	.576	.234	.241	.238	.240	.184	.170	.177	.173	.140	.147	.144	.146	
Prep. Cls.																					
None / None	No	.325	.418	.366	.395	.216	.815	.342	.525	.058	.531	.105	.202	.128	.420	.196	.288	.085	.482	.144	.249
	P1	.632	.184	.285	.214	.378	.711	.493	.604	.206	.493	.290	.385	.200	.372	.260	.317	.155	.442	.229	.322
	P2	.590	.195	.294	.226	.409	.734	.526	.633	.199	.451	.276	.360	.226	.344	.273	.311	.177	.380	.241	.309
Sentence / Chunk(200)	No	.247	.546	.340	.439	.091	.815	.164	.315	.066	.563	.119	.225	.150	.331	.206	.266	.119	.424	.186	.281
	P1	.327	.297	.311	.302	.136	.581	.220	.351	.152	.451	.227	.324	.198	.233	.214	.225	.193	.315	.239	.280
	P2	.344	.289	.314	.299	.140	.617	.228	.366	.143	.465	.219	.321	.216	.194	.205	.198	.217	.281	.245	.265
	+Art.	.333	.336	.335	.336	.138	.662	.229	.377	.153	.517	.236	.350	.225	.281	.250	.267	.199	.351	.254	.305
Sentence / Method	No	.327	.541	.408	.479	.073	.597	.130	.245	.063	.598	.114	.221	.132	.282	.180	.230	.101	.369	.159	.241
	P1	.475	.351	.403	.370	.107	.412	.169	.262	.184	.503	.269	.373	.215	.208	.211	.209	.203	.276	.234	.257
	P2	.479	.379	.423	.396	.107	.435	.172	.270	.171	.517	.257	.368	.209	.211	.210	.211	.197	.258	.224	.243
	+Art.	.486	.343	.402	.364	.109	.484	.178	.287	.168	.535	.256	.372	.203	.249	.224	.238	.190	.323	.239	.283

对于**RQ4 (RAG vs. 检索)**: 结果证实，与仅使用检索步骤的基线相比，引入LLM进行分类的完整RAG流程，在所有被测的TLR任务中，其平均F1和F2分数均有显著提升。这证明了LLM在过滤由检索步骤产生的候选链接方面的关键价值。

研究结论

- **RAG在TLR领域潜力巨大:** 本研究成功设计并验证了一个基于RAG的通用TLR框架LiSSA。实证结果清晰地表明, RAG方法在处理多样化的TLR任务时具备强大的能力, 尤其是在代码相关的追溯场景中, 其性能已能超越高度优化的专用工具。这为利用LLM解决传统软件工程难题开辟了新的研究方向。
- **当前性能尚未达到完全自动化水平:** 尽管LiSSA在多个场景下超越了基线, 但其整体性能(尤其是在F2分数所反映的召回率方面)尚未达到能够在实践中实现全自动、高可信度TLR的水平。
- **检索是核心瓶颈, 也是未来优化的关键:** 研究发现, 检索阶段的质量直接决定了整个方法性能的上限(即召回率天花板)。特别是在大型项目中, 一个文档句子可能关联数百个代码文件, 这给Top-k的检索方式带来了巨大挑战, 导致性能下降。因此, 改进检索技术, 如采用动态阈值、更先进的检索结构或工件摘要技术, 是未来工作的重中之重。
- **方法论贡献与未来方向:** 本研究不仅提出了一个可扩展的LiSSA框架, 还系统地探究了提示技术、预处理粒度等因素对性能的影响。研究指明了未来的优化方向, 包括探索更高级的RAG技术、改进聚合算法以及将LLM的推理过程融入决策中等。同时, 研究成果以可复现包的形式完全公开, 为后续研究者提供了坚实的基础。

启发: In the era of LLMs, the traceability problem is still challenging. 需要对 $M \times N$ 个(source artifact, target artifact) pairs进行二分类, 并且更加强调召回率, 即 F_2 。Benchmark数据集也很有限。

1. 标题

代码梯度：迈向 LLM 生成代码的自动化可追踪性

2. 作者及单位

Marc North; Amir Atapour-Abarghouei; Nelly Bencomo

- 杜伦大学计算机科学系，英国杜伦

- Method Grid, 英国巴斯

3. 文献来源

North, Marc, Amir Atapour-Abarghouei, and Nelly Bencomo. "*Code gradients: Towards automated traceability of llm-generated code.*" 2024 IEEE 32nd International Requirements Engineering Conference (RE). IEEE, 2024.

4. 文献类型与关键词

文章类型：本文属于一项研究预览（research preview），通过一个概念验证性实验（proof-of-concept experiment），提出并初步验证了一种新方法，因此可归类为方法学研究与实证研究。

英文关键词：Requirements Engineering; Large Language Models; Traceability

中文关键词：需求工程；大语言模型；可追溯性

研究动机

当前大型语言模型（LLM）已能在自然语言提示下生成可运行代码，但其“黑箱”性质导致需求-代码可追踪性缺失。手动维护追踪链耗时且易错，尤其在安全关键领域被视为强制要求。作者注意到：LLM 在生成代码时本质上执行“从需求到代码”的连续映射，如果能揭示这一映射，就可能在生成瞬间获得自动化追踪链，而不依赖额外测试或人工分析。

研究目标

- 利用可解释性技术实时追踪 LLM 生成代码时各输入需求对输出代码的影响；
- 自动发现并修复被模型忽视、导致需求未满足的代码缺陷；
- 通过初步实验验证梯度信息在需求-代码追踪和代码质量提升中的可行性。

研究问题

- RQ1: LLM输出相对于其输入中代表某项软件需求的部分所计算出的梯度值，如果该梯度值较低，是否与该需求在最终生成的代码中未被满足存在相关性？
- RQ2: 如果这类未被满足的需求能够被成功识别，并作为后续输入反馈给LLM，模型能否利用这些信息来修复其自己生成的代码？

Developer: Write a python function called 'add'

AI: `def add...`

- (a) Beginning of a response from CodeLlama LLM to an instruction to write a function.

Write a python function called 'add'

- (b) Visualisation of the gradients of the LLM's output with respect to each input token while generating the token 'add'.

Fig. 1: The gradients of an LLM output with respect to each input token as it generates code.

研究方法

本研究提出了一种名为“基于梯度的重提示 (gradient-based reprompting)”的方法，其核心流程如下：

- 需求分段 (Requirement Segmentation): 首先，将自然语言描述的软件需求手动拆分为代表单个、离散需求的片段。
- 梯度计算 (Gradient Calculation): 指示LLM根据完整需求生成代码。在模型生成输出的每个词元 (token) 时，计算输出相对于每个输入词元嵌入 (embedding) 的梯度。该梯度的大小反映了每个输入词元对生成该输出词元的影响力。
- 识别被忽略的需求 (Identifying Ignored Requirements): 将每个需求片段内所有词元的梯度值进行池化 (pooling)，以量化模型在生成整个代码段时对该需求片段的“关注度”。研究比较了四种池化策略 (平均值、剪枝平均值、最大值、剪枝最大值)。具有最低池化梯度值的需求片段被假定为被模型“忽略”的需求。
- 重提示修正 (Reprompting): 将识别出的被忽略需求，通过一个新的提示 (prompt) 反馈给LLM，指示模型在重写代码时“要更密切地关注”这一特定需求。
- 评估 (Evaluation): 该方法在HumanEval基准测试集上进行评估，使用了四种不同规模的LLM (WizardCoder和CodeLlama Instruct)。评估指标为代码一次性通过测试的比例 (pass@1)，并分析了代码从失败到通过的“修复 (fixes)”数量和从通过到失败的“回归 (regressions)”数量。

Prompt 1: KISS

Question: Here are two parts of software development artifacts.

{source_type}: "{source_content}"

{target_type}: "{target_content}"

Are they related?

Answer with 'yes' or 'no'.

Prompt 2: Chain-of-thought

Below are two artifacts from the same software system. Is there a traceability link between (1) and (2)? Give your reasoning and then answer with 'yes' or 'no' enclosed in <trace> </trace>.

(1) {source_type}: "{source_content}"

(2) {target_type}: "{target_content}"

研究结果

- 总体有效性:** 与模型的初次尝试相比, 采用基于梯度的重提示技术后, 代码的pass@1得分在大多数情况下得到了提升。
- 策略对比:** 在不同的梯度池化策略中, 最大值池化(max pooling)策略带来的性能提升最为显著。这可能表明, 需求片段中的少数关键语义词元对模型决策起主导作用。

Model	Model only		Max pooling		Max pooling w/ z-score		Ave pooling		Ave pooling w/ z-score		Random	
	Pass@1		Pass@1	±	Pass@1	±	Pass@1	±	Pass@1	±	Pass@1	±
WizardCoder 1b	3.0		5.5	+2.5	4.3	+1.3	5.5	+2.5	4.3	+1.3	5.5	+2.5
WizardCoder 3b	13.4		15.9	+2.5	16.5	+3.1	15.2	+1.8	15.9	+2.5	14.0	+0.6
CodeLlama 7b Instruct hf	42.7		43.9	+1.2	42.1	-0.6	42.7	+0.0	40.9	-1.8	39.6	-3.1
CodeLlama 13b Instruct hf	43.3		43.9	+0.6	44.5	+1.2	43.9	+0.6	43.9	+0.6	43.3	+0.0

TABLE I: Pass@1 scores and improvement.

- 代码回归现象:**一个重要的发现是,该方法可能导致“代码回归(code regression)”。即当模型被指示去“修复”一段本已正确的代码时,它可能会过度关注某个已满足的需求,从而引入新的缺陷,导致原先正确的代码反而无法通过测试。

Model	Max pooling		Max pooling w/ z-score		Ave pooling		Ave pooling w/ z-score		Random		
	Fixes	Regressions	Fixes	Regressions	±	Fixes	Regressions	Fixes	Regressions	Fixes	Regressions
WizardCoder 1b	8	4	6	4		8	4	6	4	7	6
WizardCoder 3b	7	3	9	4		7	4	7	3	8	6
CodeLlama 7b Instruct hf	5	3	7	8		4	4	7	10	8	13
CodeLlama 13b Instruct hf	4	3	5	3		4	3	4	3	4	4

TABLE II: The number of fixes and regressions.

- 随机基线对比：研究发现，即使是随机选择一个需求片段进行重提示，也能在一定程度上提升代码质量。这凸显了设立强力基线的重要性，以验证性能提升确实来源于基于梯度的智能选择，而非仅仅是重提示行为本身。

Model	Baseline Pass@1	Max pooling Pass@1		Max pooling w/ z-score Pass@1		Ave pooling Pass@1		Ave pooling w/ z-score Pass@1		Random Pass@1	
		±	±	±	±	±	±	±	±	±	±
WizardCode 1b	3.0	7.9	+4.9	6.7	+3.7	7.9	+4.9	6.7	+3.7	7.9	+4.9
WizardCode 3b	13.4	17.7	+4.3	18.9	+6.5	17.7	+4.3	17.7	+4.3	17.7	+4.3
CodeLlama 7b Instruct hf	42.7	47.0	+4.3	47.0	+4.3	47.0	+4.3	47.0	+4.3	47.6	+4.9
CodeLlama 13b Instruct hf	43.3	45.7	+2.4	46.3	+3.0	43.9	+0.6	43.9	+0.6	43.3	+0.0

TABLE III: Pass@1 scores and improvement over the model-only score when only rerompting the failed problems.

- 数据局限性：作者强调，由于HumanEval数据集规模相对较小，实验结果的方差较大，因此对于研究结果的普适性应持谨慎乐观态度，需要更大规模的数据集来进一步验证。

研究结论：本文首次证明输出梯度可用作需求满足程度的近似指标；基于最低梯度自动重提示，可在无需执行代码或人工审查的情况下提升LLM生成代码的正确性，实现生成时的初步需求-代码追踪。尽管数据集规模和模型多样性仍限制结论的外推性，研究为“自动化可追踪与自修复的LLM编程”开辟了新方向，并提出了更大数据集验证、梯度阈值设定及注意力机制追踪等未来工作路线。

启发：这是一项有意思的探索工作，the research problem is interesting.

1. 标题

TVR：通过检索增强生成技术实现汽车系统需求的可追溯性验证与恢复

2. 作者及单位

Feifei Niu, Hanyang Hu, Rongqi Pan, Lionel C. Briand, Krishna Koravadi

- 渥太华大学，加拿大渥太华；风河公司，加拿大；Lero中心，利默里克大学，爱尔兰；安波福，美国

3. 文献来源

Niu, Feifei, et al. “TVR: Automotive System Requirement Traceability Validation and Recovery Through Retrieval-Augmented Generation.” arXiv preprint arXiv:2504.15427 (2025).

4. 文献类型与关键词

文章类型：本文提出了一种名为TVR的新方法，并通过在真实的工业数据集上进行实证评估来验证其有效性，因此属于方法学研究与实证研究。

英文关键词：Requirements Traceability; Traceability Validation; Traceability Recovery; Large Language Models (LLMs); Retrieval-Augmented Generation (RAG); Automotive Systems

中文关键词：需求可追溯性；可追溯性验证；可追溯性恢复；大语言模型（LLM）；检索增强生成（RAG）；汽车系统

研究动机

本研究的动机源于汽车软件开发中一个紧迫的工业需求：确保利益相关者需求（stakeholder requirements）与系统需求（system requirements）之间追溯关系的正确性与完整性。

- 工业挑战：在汽车行业，由于需求变更传递不当或人工映射错误，经常出现错误或缺失的可追溯性链接，这会导致系统不一致、增加维护成本，并影响合规性。
- 现有方法不足：现有的研究方法未能有效解决利益相关者需求与系统需求之间的追溯问题，且大多依赖开源数据而非真实的工业数据，也未涉及对工程师已建立的手动链接进行验证。
- 数据复杂性：汽车领域的需求数量（见图1）在表述上存在多种多样的变体（variations），这对需要训练的监督模型构成了挑战。同时，系统需求与利益相关者需求之间的关系并非简单的文本相似性匹配，一个系统需求可能追溯到多个利益相关者需求（见图2），这使得传统方法难以胜任。
- 技术机遇：大语言模型（LLM）在自然语言理解方面的最新进展为应对上述挑战提供了新的技术机遇。

 Stakeholder Requirements

VARIATION 1 If *Trigger_Condition* = "RUN", and the module *M* does NOT receive the message *MESSAGE_1* for a certain number of message cycles defined in the "*Reference_Document*", then the module *M* shall:

Set the DTC to "Not Present" according to the appropriate validation rules contained in the "*Reference_Document*".

VARIATION 2 If *Trigger_Condition* = "RUN", the internal signal *MODULE_MODE* != "SIGNAL_1_Fail", then the module *M* must:

Set the DTC to "Not Present" according to the appropriate validation rules contained in the "*Reference_Document*".

VARIATION 3 If *Trigger_Condition* = "RUN", and the module *M* does not detect a Plausibility Fault on a signal within the message "*MESSAGE_2*", then the module *M* must:

Set the DTC to "Not Present" according to the appropriate validation rules contained in the "*Reference_Document*".

VARIATION 4 If *Trigger_Condition* = "RUN", the module *M* determines there is a failure in *SIGNAL_2*, then the module *M* shall:

Set the *INTERNAL_FLAG* = "Faulted".

Set the DTC to "Present" according to the appropriate validation rules contained in the "*Reference_Document*".

Figure 1: Fictitious Variations of Stakeholder Requirements

 System Requirement

```

if ( ENABLE_COMPONENT is enabled ) {
    if ( Missing_Msg_MESSAGE_1 || 
        Missing_Msg_MESSAGE_2 || 
        ... ) {
        LostComm_Module_M = TRUE;
    }
}

```

Figure 2: Example of Sanitized DTC System Requirement

研究目标

基于上述动机，本研究的核心目标是提出并验证一个专为汽车系统设计的、基于检索增强生成（RAG）的LLM方法，名为TVR（Traceability Validation and Recovery）。该目标具体分解为：

- 高精度验证：设计一个能高精度验证现有可追溯性链接有效性的自动化工具。
- 有效恢复：实现一个能够准确恢复缺失的可追溯性链接的解决方案。
- 鲁棒性：确保该方法对于未来可能出现的未知需求表述变体具有强大的鲁棒性与泛化能力。

研究问题

- **RQ1:** 在零样本（zero-shot）提示下，当前先进的（SOTA）LLM在需求可追溯性验证任务上的性能如何？
- **RQ2:** 针对需求可追溯性验证任务，使用性能最佳的LLM时，哪种提示策略（prompt strategy）效果最好？
- **RQ3:** 在存在未见过的需求变体时，TVR方法在需求可追溯性验证方面的鲁棒性如何？
- **RQ4:** TVR方法在恢复利益相关者需求与系统需求之间缺失的可追溯性链接方面的性能如何？

研究方法

本研究提出的TVR方法是一个基于检索增强生成（RAG）的框架（见图3），主要包含检索器（Retriever）和生成器（Generator）两个核心组件。

1. 检索器：

- 该组件首先使用Amazon Titan模型为已有验证标签的需求对生成嵌入向量。
- 当需要验证一个新的需求对时，检索器利用FAISS库计算余弦相似度，从数据库中检索出 k 个最相似的肯定（valid）示例和 k 个最相似的否定（invalid）示例。

2. 生成器：

- 该组件采用一个强大的LLM（实验证明Claude 3.5 Sonnet效果最佳）进行判断。
- 检索器获取的 $k \times 2$ 个示例被整合进一个精心设计的提示（prompt）中，作为上下文信息（见图4）。
- 生成器依据这些上下文示例分析当前待验证的需求对，并输出“Yes”或“No”的判断结果，以表明其追溯关系是否有效。

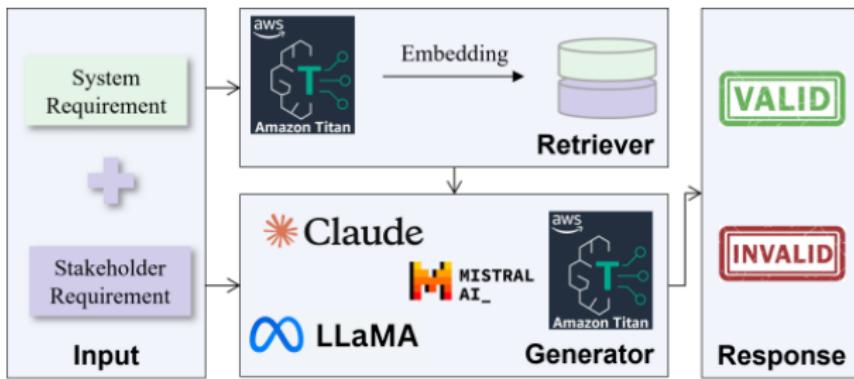


Figure 3: Overall Framework of TVR.

Input Prompt

Line 1: Please check if the message or signal from the stakeholder requirement is correctly **covered** by the system requirement.

Line 2: Please focus only on verifying the **message** or **signal** mentioned, without considering other parts of the requirement.

Line 3: Example: <example>{*k* × 2 examples}</example>

Line 4: Now evaluate the following step by step and only respond with either "Yes" or "No":

Line 5: Stakeholder Requirement: <stakeholder> {stakeReq}</stakeholder>

Line 6: System requirement: <system> {sysReq}</system>

Line 7: Response:

Figure 4: Input Prompt.

3. 可追溯性恢复:

- 为了恢复缺失链接，研究首先采用一个三步预过滤流程（变体匹配、条件匹配、消息重叠匹配）来大幅减少需要分析的候选需求对数量。
- 随后，利用TVR方法对过滤后的候选对进行验证，判定为“Yes”的即被视为需要恢复的缺失链接。

4. 评估:

- 研究使用了来自工业合作伙伴的真实汽车诊断故障码（DTC）需求数据集，包含1,320个原子利益相关者需求和48个系统需求，构成了2,132条已手动标注的追溯链接。
- 研究系统性地评估了13个SOTA LLM以及5种不同的提示工程策略（零样本、思维链、少样本、自洽性及RAG）的性能。

研究结果

- RQ1结果：在零样本提示下，所有被测LLM的性能均不理想，尤其是在识别无效链接方面表现不佳（见表1）。
- RQ2结果：逐步增强的提示策略（如思维链、少样本）能有效提升模型性能。最终，基于RAG的TVR方法表现最佳，其验证准确率达到了98.87%。实验同时确定了最佳配置为使用Claude 3.5 Sonnet模型，并结合余弦相似度检索3个示例（K=3）（见图5, 表1）。
- RQ3结果：在面对未见过的需求变体时，TVR表现出强大的鲁棒性，在跨变体验证中总体准确率仍高达97.13%（见表2）。
- RQ4结果：在可追溯性恢复任务中，经过高效的三步预过滤后，TVR成功识别出502条缺失链接，经人工验证其正确率（Correctness）为85.50%，证明了该方法在恢复任务上的有效性和实用性。

Table 1: Experimental Results of SOTA Models with Different Prompting Strategies.

Prompt	Model	Acc(%)	Valid			Invalid		
			Pre(%)	Recall(%)	F(%)	Pre(%)	Recall(%)	F(%)
Zero-shot	Claude 3.5 Sonnet	79.55	93.42	83.25	88.04	21.98	44.61	29.45
	Claude 3 Sonnet	32.08	96.51	25.83	40.75	11.51	91.18	20.44
	Claude 2	90.57	90.64	99.90	95.04	71.43	2.45	4.74
	Claude Instant	84.43	90.22	92.84	91.51	6.76	4.90	5.68
	Claude 3 Haiku	49.72	91.00	49.27	63.93	10.11	53.92	17.03
	Llama 3 8B	72.70	91.09	77.39	83.68	11.74	28.43	16.62
	Llama 3 70B	10.19	72.00	0.95	1.88	9.44	96.53	17.20
	Mistral 7B	57.09	91.96	57.45	70.72	12.08	53.77	19.72
	Mixtral 8x7B	17.23	89.76	9.62	17.38	9.43	89.55	17.07
	Mistral Large 2402	10.23	100.00	0.73	1.44	9.63	100.00	17.57
	Titan Text Premier	20.68	92.17	13.43	23.45	9.83	89.22	17.71
	Titan Text Express	49.81	90.17	49.95	64.29	9.30	48.53	15.62
	Titan Text Lite	70.83	91.49	74.69	82.24	12.54	34.31	18.37
CoT	Claude 3.5 Sonnet	76.31	98.70 ↑	74.79	85.10	27.57 ↑	90.69 ↑	42.29 ↑
	Claude 3 Sonnet	22.92	95.24	15.60	26.81	10.33	92.57 ↑	18.59
	Claude 2	66.47	94.39 ↑	66.79	78.22	17.24	63.54 ↑	27.12 ↑
	Claude Instant	86.16 ↑	90.36 ↑	94.81 ↑	92.53 ↑	8.26 ↑	4.41	5.75 ↑
	Claude 3 Haiku	45.55 ↑	94.97 ↑	41.62	57.88	13.44 ↑	80.42 ↑	23.03 ↑
	Llama 3 8B	86.25 ↑	90.56	94.73 ↑	92.59 ↑	6.00	3.30	4.26
	Llama 3 70B	35.32 ↑	98.17 ↑	28.94 ↑	44.70 ↑	12.51 ↑	94.97	22.11 ↑
	Mistral 7B	73.15 ↑	92.35 ↑	76.63 ↑	83.76 ↑	15.65 ↑	40.59	22.59 ↑
	Mixtral 8x7B	14.53	96.43 ↑	5.80	10.94	9.82 ↑	97.95 ↑	17.84 ↑
	Mistral Large 2402	10.27 ↑	100.00	0.78 ↑	1.54 ↑	9.64 ↑	100.00	17.58 ↑
	Titan Text Premier	25.70 ↑	92.57 ↑	19.40 ↑	32.08 ↑	10.07 ↑	85.29	18.01 ↑
	Titan Text Express	54.41 ↑	90.78 ↑	55.19 ↑	68.65 ↑	10.00 ↑	47.06	16.49 ↑
	Titan Text Lite	69.93 ↑	90.70 ↑	74.38	81.73	10.34	27.94	15.10
Few-shot	Claude 3.5 Sonnet	97.47 ↑	98.70	98.50 ↑	98.60 ↑	86.06 ↑	87.75	86.89 ↑
	Claude 3 Sonnet	90.81 ↑	94.00	95.95 ↑	94.97 ↑	52.44 ↑	42.16	46.74 ↑
	Claude 2	41.44	97.88 ↑	36.01	52.66	13.29	92.65 ↑	23.25
	Claude Instant	90.95 ↑	96.82 ↑	93.05	94.90 ↑	51.97 ↑	71.08 ↑	60.04 ↑
	Claude 3 Haiku	92.73 ↑	93.65	98.65 ↑	96.08 ↑	74.26 ↑	36.76	49.18 ↑
	Mistral 7B	90.01 ↑	90.62	99.22 ↑	94.73 ↑	28.57 ↑	2.94	5.33
	Mixtral 8x7B	30.66 ↑	99.13 ↑	23.56 ↑	38.07 ↑	11.90 ↑	98.03 ↑	21.23 ↑
	Mistral Large 2402	61.35 ↑	98.94	57.88 ↑	73.04 ↑	19.12 ↑	94.12	31.79 ↑
	Self-Consistency	Claude 3.5 Sonnet	97.61 ↑	98.4	98.96 ↑	98.68 ↑	89.64 ↑	84.8
	TVR (RAG)	Claude 3.5 Sonnet	98.87 ↑	99.28 ↑	99.48 ↑	99.38 ↑	95.00 ↑	93.14 ↑

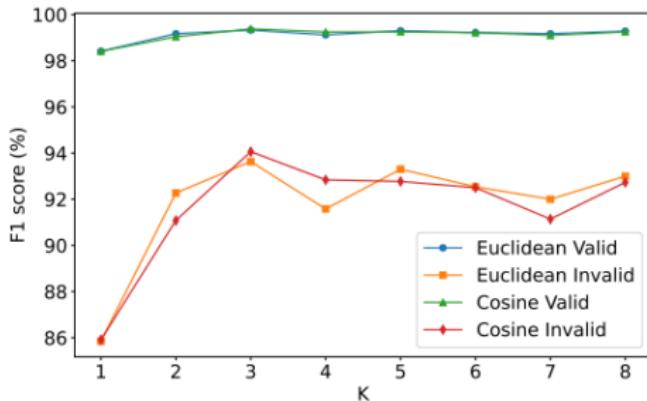


Figure 5: F1 Score Comparison for Euclidean and Cosine Similarity Across Different Values of K.

Acc(%)	Valid			Invalid			
	Pre(%)	Recall(%)	F(%)	Pre(%)	Recall(%)	F(%)	
All	97.13	97.40	99.48	98.43	93.83	74.88	83.29
V1	93.96	94.37	98.75	96.51	90.79	67.65	77.53
V2	98.78	98.81	99.89	99.35	98.28	83.82	90.48
V3	98.44	98.59	99.76	99.18	95.24	76.92	85.11
V4	92.59	95.00	95.00	95.00	85.71	85.71	85.71

Table 2: Robustness Evaluation Results.

研究结论

本研究成功证明，所提出的TVR方法，一种融合了检索增强生成（RAG）的LLM解决方案，能够高效且准确地解决汽车工业中利益相关者需求与系统需求之间的可追溯性验证与恢复问题。

- **有效性:** TVR在验证任务中取得了98.87%的准确率，在恢复任务中也达到了85.50%正确率，这些指标充分证明了其在真实工业数据上的有效性。
- **鲁棒性:** 该方法在处理未见过的需求表述变体时依然保持了高水平的准确度（97.13%），表明其具有良好的泛化能力。
- **实用价值:** TVR作为一个自动化、高精度的工具，具备在汽车工业及其他有类似需求的领域中推广应用的潜力，能够显著节约时间和成本，提升软件开发质量。

启发: 这是使用非公开工业界数据做的论文，是Lionel组的论文，可以学习/借鉴一下方法和写作(研究)范式。特点就是，方法不一定很难，但是过程严谨、规范，写作呈现很清晰。

1. 标题

论需求异味在提示中的影响：以自动化可追溯性为例

2. 作者及单位

Andreas Vogelsang, Alexander Korn, Giovanna Broccia, Alessio Ferrari, Jannik Fischbach, Chetan Arora

- 德国科隆大学、意大利国家研究委员会信息科学与技术研究所（CNR-ISTI）、爱尔兰都柏林大学、德国慕尼黑Netlight咨询有限公司、德国慕尼黑fortiss有限公司、澳大利亚蒙纳士大学

3. 文献来源

Vogelsang, Andreas, et al. “*On the Impact of Requirements Smells in Prompts: The Case of Automated Traceability.*” ICSE 2025 New Ideas and Emerging Results (NIER).

4. 文献类型与关键词

文章类型：实证研究（基准实验评估 LLM 追踪性能）

英文关键词：LLMs; Requirements Engineering; Smells; Traceability

中文关键词：大型语言模型；需求工程；异味；可追踪性

研究动机

自动化需求-代码追踪已证明大型语言模型（LLM）在高质量输入下可取得优异表现，但工业实践表明，大量需求语句天然带有“异味（smell）”一如主观词、被动语态、语义歧义等，这些质量缺陷可能削弱 LLM 的推理能力。然而，学界尚未系统量化“异味”对 LLM 追踪性能的影响，本研究旨在填补该空白。

研究目标

围绕上述痛点，作者设定三项目标：

- Benchmark：在统一数据集与度量框架下，评估主流 LLM 在“无异味”与“含异味”场景中的追踪表现；
- 量化影响：刻画异味比例及类别对追踪准确度的影响幅度；
- 实践指引：为需求规范编写与提示工程提供数据驱动的改进建议。

研究问题

- RQ1：LLM 在高质量（零异味）需求上的追踪能力如何？
- RQ2：随着需求异味比例增加，追踪性能如何变化？
- RQ3：不同异味类别（词汇、句法、语义）对追踪性能的影响是否存在显著差异？

研究方法

- **研究对象构建:** 研究基于5个游戏项目, 共包含94条需求和809行Java代码(表1)。研究者为其中的72条需求手动创建了含有“异味”的版本, 异味类型涵盖了词汇、句法和语义三大类别(表2)。
- **基准真相建立:** 由两位作者独立对无异味和有异味的需求进行审查, 以确定哪些需求被代码实现及其具体行号(LOC), 并通过讨论解决分歧, 从而建立了可靠的基准真相。
- **实验模型与度量:** 实验选用了GPT-4o和Llama 3.1作为代表模型。性能度量分为两个层面: 用于判断需求是否被实现的二元追溯准确率(BTA), 以及用于评估代码定位准确性的代码行(LOC)追溯的精确率、召回率和 F_1 分数。
- **实验执行:** 为了控制随机性, 将LLM的 $temperature$ 参数设为0, 每组实验运行5次后采用多数投票法确定最终结果。通过在每次运行中系统性地改变“异味”需求的比例, 来评估不同需求质量水平下的模型性能。

6. On the Impact of Requirements Smells in Prompts: The Case of Automated Traceability

TABLE I: Study Objects

Game	#req.	#implemented req.	#req. with smelly variant	#LOC
Dice	25	19	18	141
Arkanoid	19	14	15	152
Snake	14	7	11	142
Scopa	16	15	13	220
Pong	20	15	15	154
Sum	94	70	72	809

TABLE II: Number and types of studied smells (categorization based on Frattini et al. [9])

Smell categories and types	Description	Dice	Arkanoid	Snake	Scopa	Pong	Sum
Lexical smells							
subjective language	Words of which the semantics are not objectively defined, such as <i>user-friendly</i> , <i>easy to use</i> , <i>cost effective</i> , etc.	7 5	4 2	4 2	4 2	5 3	24 14
optional parts	Sentences expressing personal opinions or feelings.	1	1	1	0	2	5
weak verbs	Sentences containing optional parts, e.g., by using the words <i>possibly</i> , <i>eventually</i> , <i>if possible, if needed</i> , etc.	1	1	1	2	0	5
Syntactic smells							
vague pronouns	Pronouns that refer back to a previous part of the text for which the reference is unclear.	6 1	4 1	3 1	3 1	4 1	20 5
passive voice	Sentences using passive voice such that it is unclear who is performing a certain action.	3	2	1	2	2	10
negative phrases	Sentences containing negative modifiers (e.g., <i>not</i>) or negative expressions.	2	1	1	0	1	5
Semantic smells							
logical inconsistencies	Two requirements, which are connected to the same concepts, contradicting each other.	5 2	7 3	4 1	6 1	6 1	28 8
numerical discrepancies	Two requirements connected to the same concepts, containing inconsistent and/or contradicting numerical information.	1	1	1	1	1	5
ambiguities	Unclear/imprecise sentence parts that can be misunderstood if read by different people.	2	3	2	4	4	15
Sum		18	15	11	13	15	72

■ **关于RQ1 (高质量需求的追溯性能):** 在处理高质量 (0% 异味) 需求时, LLM在判断需求是否被实现的二元追溯任务上展现了高可靠性。其中GPT-4o的平均二元追溯准确率 (BTA) 为0.96, Llama 3.1为0.95。在更复杂的代码行 (LOC) 定位任务上, 性能虽有下降但仍在可接受范围内, 且GPT-4o的表现优于Llama 3.1, 其平均 F_1 分数为0.68, 而Llama 3.1为0.60 (见表III)。

TABLE III: Tracing performance with 0% smelly reqs.

Game	GPT-4o				Llama 3.1			
	BTA	LOC prec.	LOC rec.	F_1	BTA	LOC prec.	LOC rec.	F_1
dice	0.96	0.71	0.83	0.73	0.92	0.68	0.73	0.65
arkanoid	1.00	0.68	0.80	0.69	0.95	0.64	0.61	0.53
snake	0.93	0.61	0.75	0.63	0.93	0.49	0.67	0.53
scopa	1.00	0.74	0.84	0.74	1.00	0.64	0.70	0.63
pong	0.90	0.59	0.73	0.59	0.95	0.65	0.71	0.64
Mean	0.96	0.67	0.79	0.68	0.95	0.62	0.68	0.60

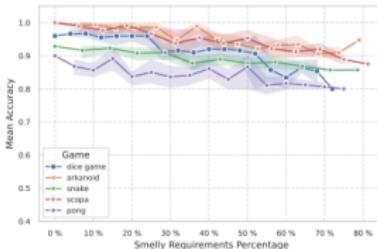
■ **关于RQ2 (异味需求的影响):** 研究采用广义线性混合模型 (GLMM) 进行量化分析, 结果显示异味需求的比例对二元追溯准确率 (BTA) 存在一个微小但统计上显著的负面影响。具体而言, 异味需求的比例每增加10%, BTA会下降0.01 ($p < 0.001$)。然而, 对于代码行追溯的 F_1 分数, 该影响在统计上并不显著 ($p = 0.055$) (见图1)。

■ **关于RQ3 (不同异味类别的影响):** 不同类别的异味对性能的影响存在差异 (见表IV)。含有句法异味 (如模糊代词、被动语态) 的需求带来的问题似乎较少。相比之下, 含有语义异味 (如不一致性、歧义) 的需求所对应的追溯性能通常最差。这一结果表明, 在使用LLM进行追溯任务时, 应优先考虑规避语义异味。

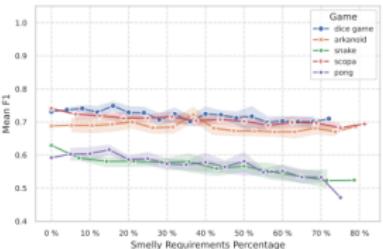
6. On the Impact of Requirements Smells in Prompts: The Case of Automated Traceability

TABLE IV: Tracing performance for smell categories

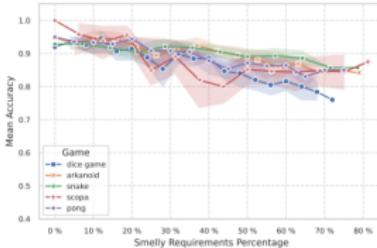
Smell category	GPT-4o				Llama 3.1			
	BTA	LOC prec.	LOC rec.	F ₁	BTA	LOC prec.	LOC rec.	F ₁
lexical	0.90	0.69	0.76	0.69	0.83	0.65	0.68	0.69
syntactic	0.98	0.74	0.84	0.73	0.91	0.76	0.75	0.73
semantic	0.83	0.63	0.76	0.63	0.86	0.64	0.70	0.63



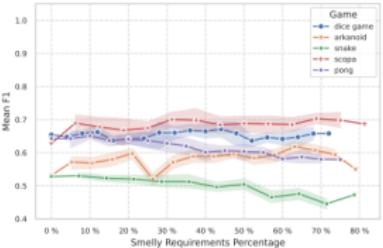
(a) Binary tracing accuracy (GPT 4o)



(b) LOC tracing F₁-score (GPT 4o)



(c) Binary tracing accuracy (Llama 3.1)



(d) LOC tracing F₁-score (Llama 3.1)

Fig. 1: Results for RQ2: Tracing performance with increasing ratio of smelly requirements.

研究结论

- 首先，在处理高质量需求时，当前先进的LLM在自动化追溯这类特定软件工程任务上表现良好，具备较高的应用潜力。
- 其次，研究表明LLM对于低质量（含有异味）的需求表现出一定的鲁棒性，尤其是在我们测试的有限复杂度的系统中，它们仍能应对部分由需求异味带来的挑战。
- 然而，这种鲁棒性并非没有边界。需求异味确实能对LLM的性能产生可观测的负面影响，尤其是在判断需求是否被实现的二元分类任务上，该影响是统计显著的。更重要的是，影响程度因异味类型而异，语义层面的问题比句法层面的问题更具破坏性。这些发现共同揭示了，深入探究需求异味在更大规模系统和其他软件工程任务中的影响，并开发相应的缓解策略，是极具价值的未来研究方向。

1. 标题

通过检索增强生成技术实现需求可追溯性链接恢复

2. 作者及单位

Tobias Hey, Dominik Fuchß, Jan Keim, and Anne Koziolek

- KASTEL信息安全与可靠性研究所，卡尔斯鲁厄理工学院（KIT），德国卡尔斯
鲁厄

3. 文献来源

Hey, Tobias, et al. “*Requirements Traceability Link Recovery via Retrieval-Augmented Generation.*” International Working Conference on Requirements Engineering: Foundation for Software Quality. Cham: Springer Nature Switzerland, 2025.

4. 文献类型与关键词

文章类型：实证研究，本文提出了一种创新的方法，并通过在六个基准数据集上的实证评估来验证其性能。

英文关键词：Traceability Link Recovery; Requirements Traceability; Requirements Engineering; LLM; Retrieval-Augmented Generation

中文关键词：可追溯性链接恢复；需求可追溯性；需求工程；大语言模型；检索增
强生成

研究动机

手工恢复需求之间的可追踪性链接既耗时又昂贵，而现有自动化方法要么性能不足，要么依赖项目已存在的训练链接，难以在缺乏先验数据的真实场景中落地。与此同时，LLM 在自然语言理解任务上表现突出，但其输入长度受限，无法一次性处理全部候选需求。检索增强生成（RAG）可先检索相关候选，再由 LLM 判断关联，为无监督需求链路恢复带来新机遇。

研究目标

- 设计一种无需初始链接的 RAG-驱动需求链路恢复方法；
- 提供首个多数据集基准，量化该方法相对传统 IR 与 SOTA 模型的效果；
- 检验 Chain-of-Thought（CoT）提示是否优于单句分类提示；
- 比较开源与商业 LLM 在任务中的可替代性。

研究问题

- RQ1: RAG-TLR 相较基线与 SOTA 方法表现如何?
- RQ2: CoT 提示能否提升性能?
- RQ3: 开源 LLM 与商业 LLM 在性能上是否可比? Requirements Traceabili…

研究方法

- 总体框架：方法包含检索与提示两步。首先用 OpenAI text-embedding-3-large 将目标需求嵌入并存入向量库；对每条源需求检索 Top-k ($k=4$) 候选，再将〈源, 候选〉对组装成提示，由 LLM 判定是否存在链路（见图1）。
- 提示策略：实现 KISS（零样例二分类）与 CoT（要求模型给出推理后再回答）两种提示。
- 数据集：六个公开基准（CCHIT、CM1、Dronology、GANNT、MODIS、WARC），涵盖 R-RC 及 HLR-LLR 场景，共 336 件源需求与 1 535 件目标需求，金标准链接总计 1 097 条（见表1）。
- 模型与评价：GPT-4o / GPT-4o mini、CodeLlama-13B、Llama 3.1-8B；指标采用 Precision、Recall、F1 与 F2；IR-only、VSM、LSI 为基线，另对比 WQI、S2Trace、Task-CLS/RANK。实验均设温度 0、固定随机种子 133742243，保证可复现。

7. Requirements Traceability Link Recovery via Retrieval-Augmented Generation

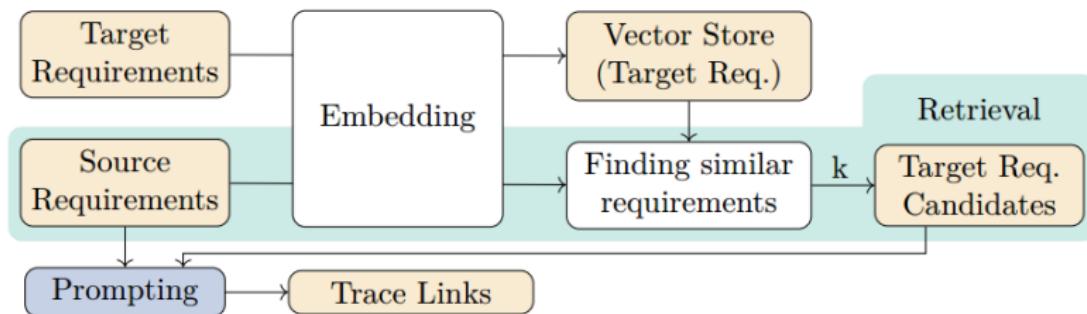


Fig. 1. Overview of the retrieval and prompting steps with data in orange, prompting in blue, and the other processing in white.

Table 1. Overview of the datasets. Datasets comprise either high-level requirements (HLR), low-level requirements (LLR), requirements (R) or regulatory codes (RC). Percentage of linked source and target artifacts in the gold standard is given in brackets.

Dataset	Artifact Type		Number of Artifacts		
	Source	Target	Source	Target	TLs
CCHIT-2-WorldVista	R	RC	116 (62%)	1064 (39%)	587
CM1-NASA	HLR	LLR	22 (86%)	53 (57%)	45
Dronology	HLR	LLR	99 (96%)	211 (99%)	220
GANNT	HLR	LLR	17 (100%)	69 (99%)	68
MODIS	HLR	LLR	19 (63%)	49 (63%)	41
WARC	HLR	LLR	63 (95%)	89 (89%)	136

研究结果

- RQ1：在六数据集平均 F1 上，GPT-4o+CoT 以 45.1 % 领先全部基线和现有方法（见表3），除 MODIS 外均为第一；对 VSM、LSI 的提升在 $\alpha = 0.05$ 水平上显著，效应量中等。

Table 3. Comparison to baselines and existing approaches. Our approaches are underlined and the best results per column are highlighted in bold type.

Approach	CCHIT		CM1		Dronol.		GANNT		MODIS		WARC		Avg.		
	F ₁	F ₂													
WQI	—	—	.349	.337	—	—	.351	.453	—	—	—	—	—		
S2Trace	—	—	.378	.435	—	—	.381	.463	—	—	—	—	—		
Task-CLS	—	—	—	.605	—	.493	—	—	—	—	—	—	—		
Task-RANK	—	—	—	.520	—	.510	—	—	—	—	—	—	—		
Baseline	VSM	.190	.259	.429	.451	.519	.552	.344	.374	.333	.273	.221	.303	.339	.369
	LSI	.186	.203	.374	.413	.487	.503	.326	.387	.292	.307	.165	.207	.305	.337
	IR-only	.175	.164	.436	.541	.497	.600	.544	.544	.188	.229	.485	.590	.387	.445
KISS	GPT-4o	.188	.168	.446	.547	.503	.603	.544	.544	.229	.251	.496	.597	.401	.452
	Codellama	.190	.167	.446	.547	.496	.561	.544	.544	.268	.268	.487	.532	.405	.437
CoT	GPT-4o	.200	.158	.519	.565	.575	.620	.574	.556	.255	.197	.584	.616	.451	.452
	Llama3.1	.176	.145	.504	.580	.473	.526	.550	.537	.238	.242	.516	.584	.410	.436

- RQ2：除 CodeLlama 外，其余三模型使用 CoT 的平均 F1 均高于 KISS；GPT-4o+CoT 较 IR-only F1 提升 7.6 个百分点且统计显著，但 CoT 与 KISS 间差异整体未达显著水平（见表2）。

- RQ3：在 CoT 提示下，开源 CodeLlama、Llama 3.1 的 F1/F2 与 GPT-4o mini 无显著差异；唯 GPT-4o 对开源模型在 F1 上具显著优势，中等效应量，F2 无显著差异。

7. Requirements Traceability Link Recovery via Retrieval-Augmented Generation

Table 2. Detailed results of our RAG-based approach with different models and prompt types. Best results per row are highlighted in bold type.

Dataset	Metric	KISS				CoT				
		GPT-4o	GPT-4o mini	Code llama	Llama 3.1	GPT-4o	GPT-4o mini	Code llama	Llama 3.1	
CCHIT	P.	.198	.234	.212	.247	.205	.367	.264	.206	.276
	R.	.157	.157	.157	.155	.157	.138	.145	.152	.129
	F ₁	.175	.188	.180	.190	.178	.200	.187	.175	.176
	F ₂	.164	.168	.165	.167	.164	.158	.159	.160	.145
CM1	P.	.330	.341	.330	.341	.330	.458	.358	.438	.414
	R.	.644	.644	.644	.644	.644	.600	.644	.467	.644
	F ₁	.436	.446	.436	.446	.436	.519	.460	.452	.504
	F ₂	.541	.547	.541	.547	.541	.565	.556	.461	.580
Dronology	P.	.386	.394	.386	.417	.383	.512	.421	.388	.405
	R.	.695	.695	.695	.614	.686	.655	.668	.632	.568
	F ₁	.497	.503	.497	.496	.492	.575	.517	.481	.473
	F ₂	.600	.603	.600	.561	.593	.620	.598	.561	.526
GANNT	P.	.544	.544	.544	.544	.561	.607	.569	.565	.571
	R.	.544	.515	.529						
	F ₁	.544	.544	.544	.544	.552	.574	.556	.538	.550
	F ₂	.544	.544	.544	.544	.547	.556	.549	.524	.537
MODIS	P.	.145	.200	.193	.268	.220	.500	.263	.153	.233
	R.	.268	.268	.268	.268	.268	.171	.244	.220	.244
	F ₁	.188	.229	.224	.268	.242	.255	.253	.180	.238
	F ₂	.229	.251	.249	.268	.257	.197	.248	.202	.242
WARC	P.	.373	.387	.373	.428	.374	.537	.408	.405	.433
	R.	.691	.691	.691	.566	.676	.640	.654	.625	.640
	F ₁	.485	.496	.485	.487	.482	.584	.503	.491	.516
	F ₂	.590	.597	.590	.532	.582	.616	.584	.564	.584
Avg.	P.	.329	.350	.340	.374	.345	.497	.381	.359	.389
	R.	.500	.500	.500	.465	.496	.458	.483	.435	.459
	F ₁	.387	.401	.394	.405	.397	.451	.413	.386	.410
	F ₂	.445	.452	.448	.437	.447	.452	.449	.412	.436

研究结论

作者验证了 RAG 可显著提升无监督需求链路恢复性能，并在多数数据集超越传统 IR 与近期 SOTA 方法；CoT 提示在精、召同权场景中具潜在益处；当关注召回 (F2) 时，开源 LLM 足可替代商业模型；但 45 % F1 仍不足以完全自动化，需在提示工程、上下文注入与需求片段识别等方向进一步改进。

启发：这几个数据集可以增加进来，感觉有几个数据集是不眼熟的。

1. 标题

结合检索增强生成与大型语言模型的自然语言需求—软件构件可追踪性建立

2. 作者及单位

Syed Juned Ali; Varun Naganathan; Dominik Bork

- 维也纳工业大学商业信息学研究组（Business Informatics Group, TU Wien），奥地利；微软公司海得拉巴研究院，印度

3. 文献来源

Ali, Syed Juned, Varun Naganathan, and Dominik Bork. “Establishing Traceability Between Natural Language Requirements and Software Artifacts by Combining RAG and LLMs.” International Conference on Conceptual Modeling. Cham: Springer Nature Switzerland, 2024.

4. 文献类型与关键词

文章类型：方法学研究并伴随实证评估。

英文关键词：Large Language Models; Requirements Traceability; Retrieval Augmented Generation; Requirements Engineering

中文关键词：大型语言模型；需求可追踪性；检索增强生成；需求工程

研究动机

需求-代码可追踪性是维护、影响分析和再利用的前提，但手工建立链路高成本且易错。传统基于文本相似度的方法难以跨越“自然语言需求 \leftrightarrow 代码”语义鸿沟，导致低精度问题。随着源代码频繁演化，保持UML模型同步更加困难，迫切需要一种能直接在代码层面建立精确链路的自动化方案。同时，LLM虽能理解NL与代码，却受上下文长度与领域知识缺失限制；RAG为LLM注入外部上下文，被认为有望缓解上述瓶颈。

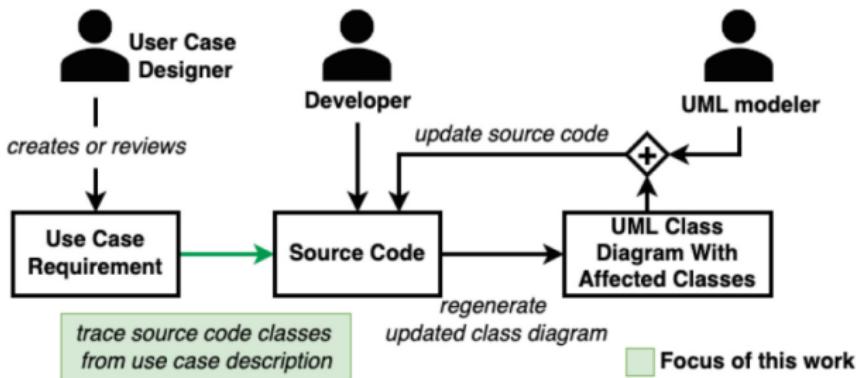


Fig. 1. Requirements to Code Traceability-based UML Classes Evolution

研究目标

- 借助关键词-向量-知识图三重索引与 LLM 代码摘要，实现高精度链路恢复；
- 系统评估不同索引/参数组合对性能的影响，为 RAG 质量优化提供指导；
- 与现有最佳方法比较，提升精度同时保证可扩展性；

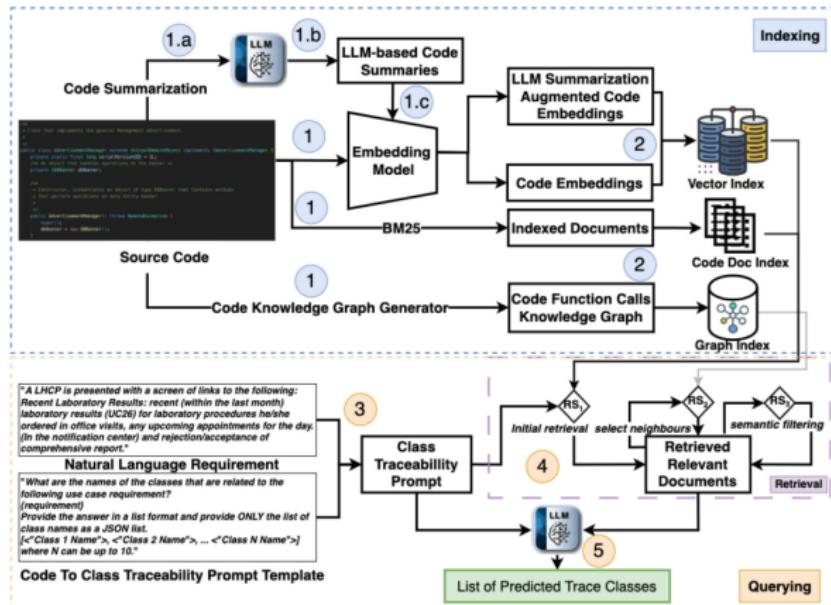


Fig. 2. Retrieval Augmented Requirements Traceability Generation

研究问题

- RQ1: RARTG 在四个公开数据集上的追踪性能相较最新工作如何？
- RQ2: RARTG 的五个核心参数—LLM 生成摘要 (GS)、方法调用显式化 (MC)、查询扩展 (QE)、知识图索引 (KGI)、多索引组合 (CI) —对性能有何影响？

研究方法

- 总体流程：如图2 所示，先对代码生成 LLM 摘要并构建 KI、VI、KGI；检索阶段通过 RS1-RS3 逐步融合语义与结构上下文，并将精选文档连同需求组成提示交由 LLM 判定相关类。
- 数据集：eTour、iTrust、SMOS、eANCI，语言涵盖 EN/IT，统计见表1（见表1）。
- 评价指标：Precision、Recall、F1；与 IR-only、VSM、LSI 及文献 [14] 提供的 SOTA 结果对比。
- 实验设计：对五参数取全组合共 $2^5 = 32$ 组配置，固定温度与随机种子确保可复现。

研究结果

RQ1 结果: 最佳配置 RARTG-C2 在 iTrust 与 eANCI 上 F1 分别提升至 0.285 与 0.287，相比基线提高约 6%（见表2）；Precision 在四数据集均显著优于基线。

RQ2 结果: 参数显著性分析见表3：

- GS 在 iTrust、eANCI 上呈显著正效应，验证了 LLM 摘要对提高语义匹配的价值；
- MC 对 eANCI、SMOS 为正，但在 iTrust 出现负面影响，显示其依赖数据集特征；
- QE 在 iTrust、eTour 显著负效，应谨慎使用；
- KGI 与 CI 的引入在多数场景产生正增益，表明结构信息与多索引融合可弥补纯语义检索的遗漏。

Table 2. Traceability comparison across evaluation datasets

Dataset	iTTrust			eANCI			SMOS			eTour		
Approach	Prc	Rec	F1									
Baseline	0.176	0.353	0.235	0.294	0.220	0.252	0.443	0.297	0.356	0.411	0.623	0.495
RARTG-C2	0.284	0.286	0.285	0.737	0.178	0.287	0.526	0.118	0.192	0.488	0.239	0.321
RARTG-C1	0.289	0.292	0.290	0.779	0.199	0.317	0.608	0.126	0.209	0.543	0.242	0.334

Prc: Precision, Rec: Recall

Table 3. Impact of parameter change on the F1-score

Parameter	iTTrust			eANCI			SMOS			eTour		
	TS	P	Effect	TS	P	Effect	TS	P	Effect	TS	P	Effect
GS	2.60	0.02	SP	5.48	<0.05	SP	0.40	0.70	-	1.60	0.13	-
MC	-2.36	0.03	SN	2.24	0.04	SP	3.22	<0.05	SP	1.78	0.09	Pos.
QE	-4.77	<0.05	SN	1.72	0.10	-	0.29	0.77	-	-3.14	0.01	SN
KGI to VI	25.84	0.02	SP	-0.31	0.77	-	0.36	0.73	-	15.75	<0.05	SP
KGI to KI	0.18	0.87	-	0.40	0.70	-	12.24	<0.05	SP	3.08	0.02	SP
CI to VI	6.34	0.00	SP	0.58	0.58	-	1.74	0.12	-	7.96	<0.05	SP
CI to KI	-0.19	0.86	-	3.42	0.01	SP	16.32	<0.05	SP	3.06	0.02	SP

SP: Significant Positive, **SN:** Significant Negative, **-:** Inconclusive,

Pos. - Positive

TS: T-statistic, **P:** p-value

研究结论

RARTG 结合 LLM 摘要与三重索引，在无需训练数据的情况下实现了对现有方法的精度领先；其中索引语义质量与结构上下文是性能提升关键（见表3）。作者认为该框架为大规模、跨语言项目的自动化可追踪性提供了可行途径，并计划通过放宽初始检索阈值与分层检索策略进一步提升召回率。

启发：在大语言模型以及RAG的加持下，traceability常用公开数据集上的performance指标还是没提升上去，这其实跟该任务的特点有关，source和target之间存在语义鸿沟，依赖project-level的上下文和software engineering background。所以呀，这个任务可不是一个simple pair classification，就好比NLP中的语义匹配、语义蕴含这些任务并不是简单的文本分类，甚至traceability更难，因为有语义鸿沟和需要project-level (dialogue-level)的背景知识。

1. 标题

基于大型语言模型的需求与 MBSE 构件可追踪性自动建立方法

2. 作者及单位

Maria Bonner; Marc Zeller; Gabor Schulz; Ana Savu

- 西门子股份公司 Siemens AG (数字化工业、技术研究院等)

3. 文献来源

Bonner, Maria, et al. “*LLM-based Approach to Automatically Establish Traceability between Requirements and MBSE.*” INCOSE International Symposium. Vol. 34. No. 1. 2024.

4. 文献类型与关键词

文章类型：方法学研究并伴随实证评估

英文关键词：LLM; Requirement Engineering; MBSE; Semantic Web; Knowledge Graph

中文关键词：大型语言模型；需求工程；基于模型系统工程；语义网；知识图谱

研究动机

汽车 E/E 系统的安全标准（ISO 26262）要求在不同层级需求与实现工件之间建立双向可追踪关系，但现实中大量异构且用自然语言描述的需求与多种 MBSE 模型使人工建立链路既耗时又易出错，导致工程师常因担忧错误链接而放弃追踪。因此，亟需一种能够自动弥合需求文本与 MBSE 模型“语义鸿沟”的高效追踪技术。

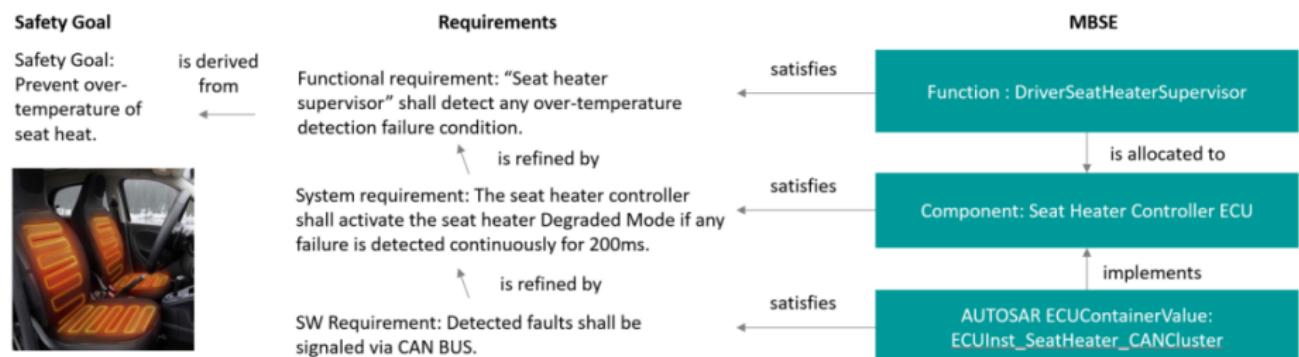


Figure 3 Example for trace links between requirements and MBSE

研究目标

基于以上痛点，本文旨在设计并验证一种结合语义网技术与大型语言模型（LLM）的追踪推荐框架，利用嵌入相似度与分类模型，自动为工程师提供高质量的需求-模型候选链路，并以工业数据集评估其有效性（见图 1）。

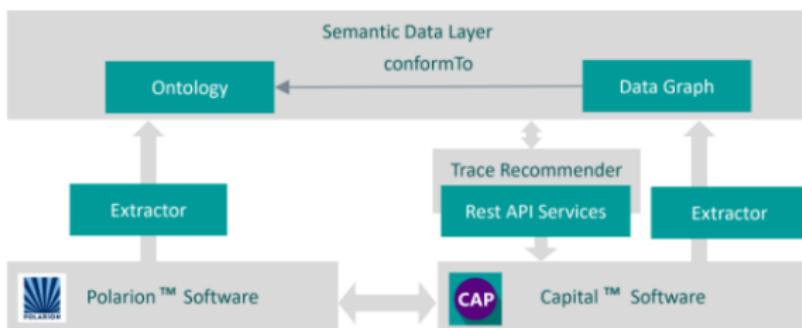


Figure 1 Architecture of the trace recommender tool.

研究问题

- RQ1: LLM 嵌入相似度能否在复杂工业数据集中自动生成高质量需求-MBSE 追踪链路?
- RQ2: 语义预处理与领域微调对 LLM 追踪性能的提升幅度如何?
- RQ3: 与经典信息检索算法 TF-IDF 相比, 微调后 LLM 的召回率、精确率及 F2 得分提升是否具有统计学意义?

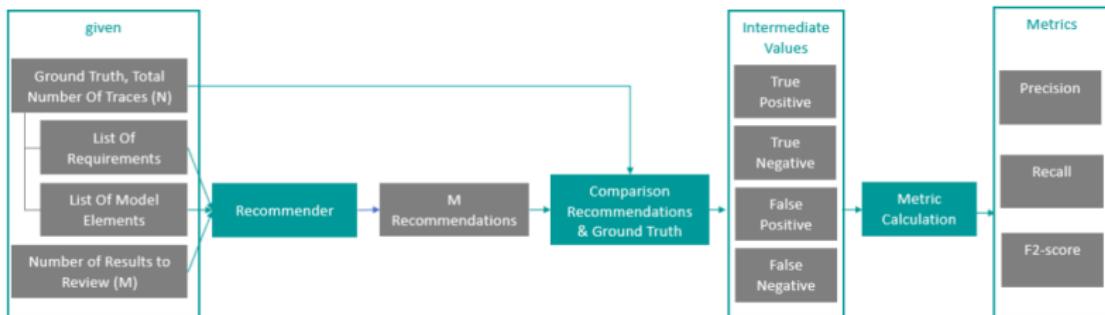


Figure 7 Evaluation algorithm for the traceability recommender.

研究方法

- 数据语义化：通过 ETL 将需求与模型导入 RDF/OWL 语义层¹，随后执行“展开”与可选的分词、词干化等处理，为 LLM 提供语境丰富的输入；
- LLM 嵌入与微调：选用 all-MiniLM-L6-v2 与 all-mpnet-base-v2 预训练模型，在工业 CDCBase 数据集上以 MultipleNegativesRankingLoss 进行五轮微调。
- 追踪推荐工具：将算法集成到 Tracer Recommender Tool，实现需求-模型候选展示、人工接受/拒绝及可视化仪表盘。
- 实验设计：采用 8 : 2 训练-测试划分，设定用户审阅列表长度 M，对比 TF-IDF、未经微调的 LLM 及微调 LLM，计算 Recall、Precision 与 F2 指标。

¹ETL (Extract, Transform, Load) 流程把“散落各处、格式各异”的需求和 MBSE 模型规范化为可机读的 RDF/OWL 知识图谱，为大语言模型提供统一、高质量的输入语境

研究结果

- 微调 LLM 在五类追踪任务中的F2得分相较 TF-IDF 提升 10%-60%，在链路稀疏场景亦保持优势。
- 在“系统安全需求→SW 安全需求”任务中，只需审阅前 10 个候选即可发现 > 90% 的真实链路，显著降低人工负担。
- 语义上下文丰富度与微调数据同构性对性能提升最为关键；跨任务迁移可能导致效果回退，需要按任务细粒度微调。

研究结论

本文验证了“语义网预处理 + LLM 微调”策略在需求-MBSE 自动追踪中的有效性：微调 LLM 明显优于传统 IR 和未微调模型，可在可接受的审阅范围内提供高召回、高精度的链路推荐，证明在工业场景投入微调资源是值得的。未来工作将针对特定领域进一步定制 LLM，并扩展到模型-模型及需求-测试用例等更多追踪场景。

启发：这篇论文的思路是微调sentencebert模型，转化为semantic match任务，用learn to rank目标函数。不直接做判别分类，因为太**简单粗暴了**，很缺乏对任务特点的考虑和建模。

1. 标题

一种基于LLM的恢复安全需求与目标模型之间可追溯性链接的方法

2. 作者及单位

Jameleddine Hassine

- 法赫德国王石油与矿产大学（KFUPM）信息与计算机科学系、法赫德国王石油与矿产大学（KFUPM）智能安全系统跨学科研究中心

3. 文献来源

Hassine, Jameleddine. "*An llm-based approach to recover traceability links between security requirements and goal models.*" Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering. 2024.

4. 文献类型与关键词

文章类型：方法学研究、实证研究。

英文关键词：Large Language Model (LLM); traceability link; security requirements; Goal-oriented Language (GRL); GPT-3.5-turbo

中文关键词：大型语言模型 (LLM); 可追溯性链接; 安全需求; 目标导向语言 (GRL); GPT-3.5-turbo

研究动机

对安全需求与目标模型（GRL）之间建立双向可追踪链路，是保证系统满足安全目标的关键；然而两类工件一方以自然语言表述，另一方以图-文本混合的意图元素表示，人工追踪既费时又易漏误。作者以“虚拟室内设计”应用的复杂 GRL 模型为例，指出现有流程难以精准定位安全相关映射。伴随 GPT-3.5 等大语言模型（LLM）在语义理解上的突破，探索其在自动追踪中的潜力成为迫切需求。

研究目标

本研究旨在设计并验证一种零样本-提示驱动的 LLM 方法，通过精心构造的系统提示，自动恢复安全需求与安全目标（含 softgoal、goal、task）的交叉追踪链路，并开发原型工具以支持实际工程使用。

- R1: The app shall allow users to set privacy preferences for their designs, controlling who can view or edit them.
- R2: The app shall provide end-to-end encryption for all user data, ensuring privacy during transmission and storage.
- R3: The app shall integrate with trusted payment gateways like PayPal or Stripe to secure financial transactions.
- R4: The app shall implement tokenization for sensitive financial information to safeguard against unauthorized access.
- R5: The app shall encrypt user data using AES-256 encryption before storing it on servers.
- R6: The app shall regularly update encryption protocols to maintain strong data security standards.
- R7: The app should optimize image and asset loading to achieve fast loading speeds, especially for 3D models.
- R8: The app should utilize content delivery networks (CDNs) to reduce latency in loading assets across different regions.
- R9: The app shall enforce password complexity requirements and provide options for multi-factor authentication.
- R10: The app shall log and monitor login attempts, automatically locking accounts after multiple failed login attempts to prevent unauthorized access.
- R11: The app shall use machine learning algorithms to detect patterns indicative of fraudulent behavior.
- R12: The app shall notify users and administrators of suspicious activities and provide guidance on how to secure their accounts.

Goal Models

1

Virtual Interior Designer App

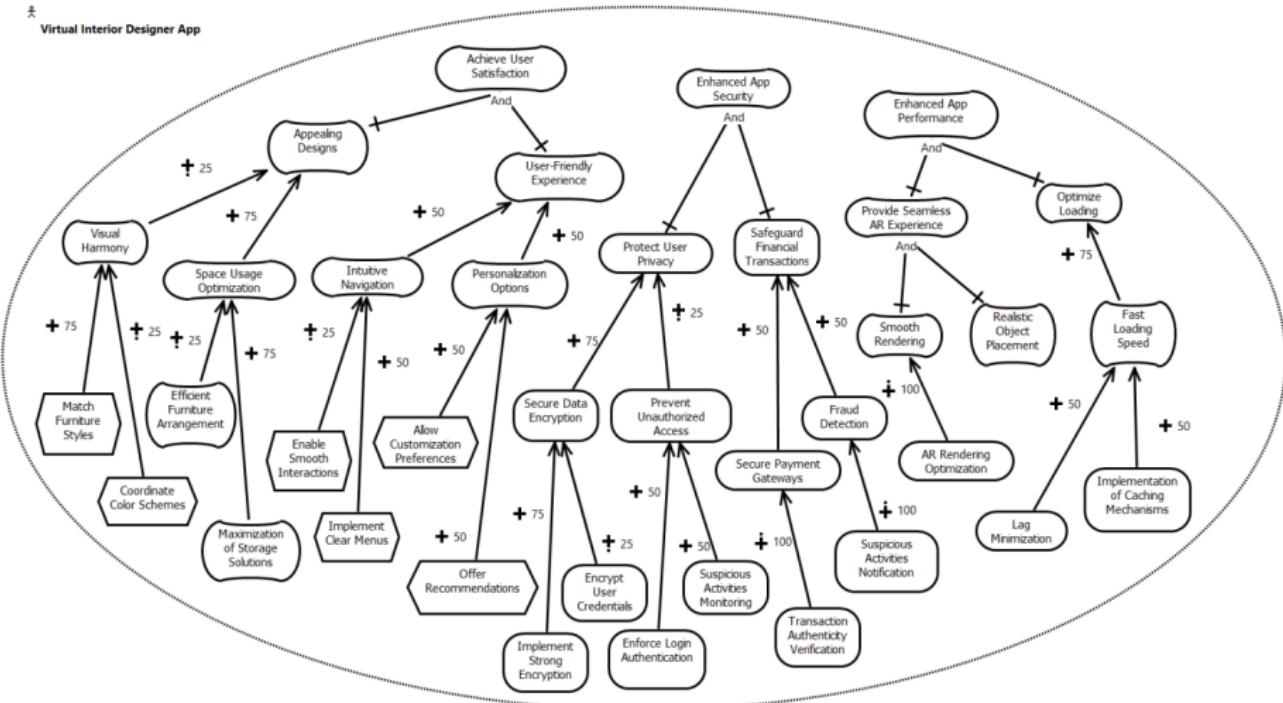


Figure 1: Virtual Interior Design App GRL model

研究问题

- RQ1: 零样本 GPT-3.5-turbo 能否在不依赖标注数据的情况下，正确识别安全需求与 GRL 安全目标之间的追踪关系？
- RQ2: 在给定的安全密集场景中，其精确率、召回率与 F₁ 得分达到何种水平，是否满足工程可用性？研究以 42 条需求与 37 个 GRL 意图元素构成的评测基准展开。

```
prompt_messages = [
    {"role": "system",
     "content": [
         f"You are a requirements engineer. Given the following requirements {Requirements} "
         f"and the following goals {goals}. "
         f"The task consists of linking the security-related requirements within {Requirements} "
         f"to the security-related goals from {goals}. "
         "Note that one goal can be linked to many requirements and vice versa."
         "Provide the rationale for each generated link separately."
         "Do not report non-security related links."},
        {"role": "user",
         "content": ""}
    ]
]
```

Figure 5: Prompt for security-related traceability link generation

研究方法

- 提示工程：采用系统角色-用户角色双通道提示，限定模型仅输出安全相关链路，并要求给出每条链路的合理性说明。
- 工具实现：基于 Gradio 实现交互式 GUI，集成文件上传、LLM 调用与结果可视化。
- 效果评估：依据 TP/FP/FN 计算 Precision、Recall 与 F1-score，人工校验链路正确性，避免偏差。

研究结果

LLM 共生成 11 条安全追踪链路，全部为真实链路 ($TP=11, FP=0$)，另有 3 条真实链路漏检 ($FN=3$)。综合指标：Precision = 100%、Recall = 78.5%、F1 = 87.9%。输出的每条链路均附带逻辑自洽的理由说明。

Table 1: Evaluation Results

Metric	Value
True Positive (TP)	11
False Positive (FP)	0
False Negative (FN)	3
Precision	1
Recall	0.785
F1-Score	0.879

6. ****Fraud Detection****
 - **Linked Requirement:**
 - R11: The app shall use machine learning algorithms to detect patterns indicative of fraudulent behavior.
 - **Rationale:**
 - Requirement R11 focuses on using machine learning algorithms to detect patterns indicative of fraudulent behavior, thereby enhancing fraud detection capabilities within the app.

 7. ****Enhanced App Security****
 - **Linked Requirement:**
 - R19: The app shall regularly undergo security audits and penetration testing to identify and address potential vulnerabilities, ensuring enhanced app security.
 - **Rationale:**
 - Requirement R19 emphasizes the importance of regular security audits and penetration testing to identify and address potential vulnerabilities, ultimately ensuring enhanced app security.
-

Figure 7: Goal-Requirement Traceability links generated output

研究结论：零样本 GPT-3.5-turbo 结合精细提示，可在无训练数据的情况下高质量恢复安全相关追踪链路，Precision 达到 100%、Recall 接近 80%，足以显著减轻工程师手工追踪负担；LLM-prompt 方案为安全需求-目标一致性分析提供了一条轻量、可复制的技术路线。

启发：单项目，小规模数据量的验证²，结果的外部效度有待提高。

² 研究中使用的GRL模型和需求文本是“由参加需求工程课程的本科生开发的”，这些特定的教学产物并未作为标准化的复现包对外发布。

1. 标题

分类还是提示：一项关于法律需求可追溯性的案例研究

2. 作者及单位

Romina Etezadi, Chetan Arora, Sallam Abualhaija, Lionel Briand

- 渥太华大学电气工程与计算机科学学院，加拿大；卢森堡大学安全、可靠性与信任中心（SnT），卢森堡；莫纳什大学信息技术学院，澳大利亚，墨尔本；Lero爱尔兰软件研究中心和利默里克大学，爱尔兰

3. 文献来源

Etezadi, Romina, et al. “*Classification or Prompting: A Case Study on Legal Requirements Traceability.*” arXiv preprint arXiv:2502.04916 (2025).

4. 文献类型与关键词

文章类型：方法学研究、实证研究。

英文关键词：Requirements Traceability; Sentence Transformers (ST); Natural Language Processing (NLP); Machine Learning (ML); The General Data Protection Regulation (GDPR); Regulatory Compliance; Large Language Models (LLMs); RICE

中文关键词：需求可追溯性；句子转换器(ST)；自然语言处理(NLP)；机器学习(ML)；通用数据保护条例(GDPR)；法规遵从性；大语言模型(LLM)；RICE

研究动机: 随着 HIPAA、GDPR 等法规持续更新, 软件产品必须证明其需求文档已满足相应法律条款。然而法规文本以“法律语言”书写, 与技术需求之间存在**显著术语鸿沟**, 且真实项目往往包含成百上千条需求, 靠人工建立 requirements-provisions 追踪链路既耗时又易遗漏。既有基于概率关键字或传统信息检索的追踪方法在召回率、跨域泛化能力方面均表现不佳(见表3, Baseline 仅取得 13.3% 召回), 亟需更先进的自动追踪技术。

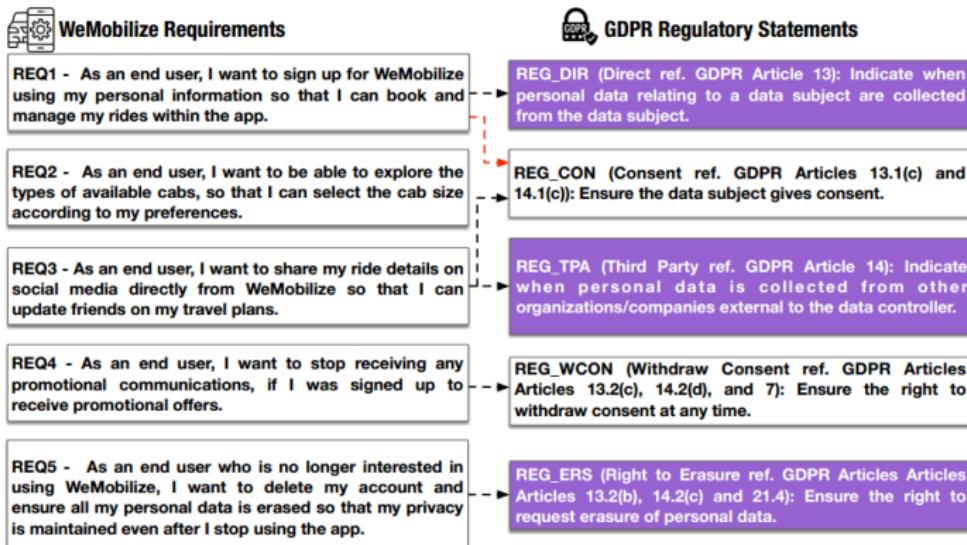


Fig. 1 Example on tracing WeMobilize app requirements to GDPR statements.

研究目标

本文旨在系统评估大语言模型（LLM）在法律需求追踪（Legal Requirements Traceability, LRT）中的有效性，具体目标为：

- 构建并调优一个基于 Sentence Transformers 的分类框架 Kashif，探测需求-法规链路；
- 设计一个基于 Rice 提示工程框架³的零样本 LLM 方案，验证其在复杂法规场景中的可行性；
- 通过基准数据集与实际 GDPR 案例对两种方案进行对比实验，给出使用建议。

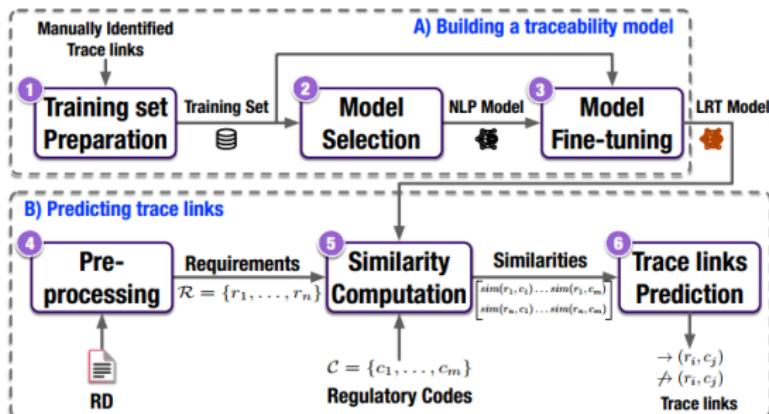


Fig. 2 Overview of Kashif.

³Role-Instruction-Context-Example-Constraint

研究问题

- RQ1: 38 个预训练 Sentence-Transformer 模型中，何者在 HIPAA 数据集上零样本表现最佳？
- RQ2: 在 HIPAA 基准上，Kashif 相较文献基线算法的精度、召回与 F1 表现如何？
- RQ3: Kashif 在未见过的多领域 GDPR 需求文档上能否保持性能？
- RQ4: Rice + GPT-4o 在同一 GDPR 数据集上的效果与 Kashif 有何差异？

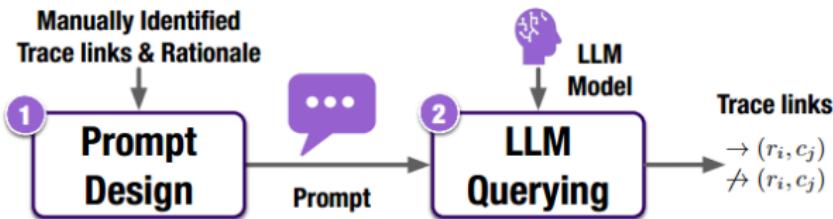


Fig. 3 Overview of RICE.

研究方法

- 数据集: 采用含 1871 条需求、10 条 HIPAA 条款的公开数据集作训练/验证, 并额外标注 4 份跨域需求文档与 26 条 GDPR 条款。
- Kashif: 选取 paraphrase-multilingual-mpnet-base-v2 (ST29) 为基础模型, 经 HIPAA 数据集微调 10 轮; 提出常数阈值、动态阈值、最大间隙阈值三种判定策略。
- Rice: 依据 Role-Instruction-Context-Example-Constraint 理念构造五段式提示, 内含示例链路与推理;
- 在 GPT-4o 上以零样本方式逐条预测并附带理由。
- 评价指标: 采用 Precision、Recall、F1 与 ROC-AUC; 对 HIPAA 使用留一交叉验证, 对 GDPR 文档按需求-条款两级统计; 并分析误差来源。

11. Classification or Prompting: A Case Study on Legal Requirements Traceability

[Context] I am currently working on a task focused on establishing traceability between software requirements and regulatory codes.¹⁰ This involves analyzing and mapping requirements to relevant GDPR regulations, ensuring that our software development aligns with regulatory compliance. Below are the main regulatory codes that I want you to remember at first: {The 26 regulatory codes with their descriptions + a 27th code capturing the "ELSE" value indicating no trace link.}

[Examples] Here are five sample traceability examples. I've also added my rationale for tracing regulatory codes to the requirements for your reference. {Five example requirements along with their trace links and the rationale behind selecting these links. Requirement: TEXT. trace links: LIST, rational behind choosing these codes: TEXT.}

[Instruction] Find the trace links for a given requirement and provide the rationale behind your choice extended from the examples I provided. Please consider regulatory codes which I have not used in the examples. Pay attention to the roles (AS_ROLE) in the requirement, if there are any. Remember, regulations' text focus on personal data, but try to consider all types of data, role, or functionalities in a software system. Pay attention to commonsense and indirect relations between requirement and regulations. Aim to include regulations even if they have a low likelihood of being traced, prioritizing recall over precision. Choose at least one regulation for each requirement.

[Output Indicator] List of alphabetical order of regulatory codes (if any) similar to the examples I provided to you. Newline to explain the rational behind the choice(s).

Note that we use regulatory codes to mean provisions, since the former was used in the literature [10]

Fig. 4 Final RICE-based prompt for addressing LRT.

GPT4o Example Output

REQ: *If a key file is required to unlock the database, the database doesn't open unless the key file is present.*

Trace links: [ACC, CNF, SEC]

Rationale:

- ACC^a: The presence of the key file is required to gain access to the database, ensuring proper authentication.
- CNF^b: Protecting the database with a key file ensures that sensitive data remains confidential and accessible only to those with the correct authentication file.
- SEC^c: Requiring a key file adds an additional layer of security, preventing unauthorized access even if the password is compromised.

^a ACC: Right to access.

^b CNF: Ensure confidentiality.

^c SEC: Ensuring security.

研究结果

- RQ1: ST29 在 HIPAA 上取得最高 AUC 0.859。
- RQ2: Kashif-constant 的 F1 为 56.9%，较基线提升 35.3 个百分点，召回率由 13.3% 升至 67.3%。
- RQ3: 当直接迁移到 GDPR，Kashif 召回骤降至 15.0%，显示分类式方案缺乏跨法规泛化能力。
- RQ4: Rice + GPT-4o 在 GDPR 文档上平均召回 84.0%，成功率（Exact+Partial）超 90%，显著优于 Kashif；多数需求仅需人工核查 1-3 条候选链路。

Table 3 Accuracy of *Kashif* and B on HIPAA (RQ2).

	TP	FP	FN	P	R	F1
<i>Kashif</i> constant	111±12	114±8	54±4	49.3±13.2	67.3±18.5	56.9±12.3
<i>Kashif</i> dynamic	122±12	441±46	43±4	21.7±18.5	73.9±21.3	33.5±13.6
<i>Kashif</i> Δ	132±12	1531±81	33±2	7.9±3.4	80.0±12.5	14.4±5.8
B	22±2	16±1	143±10	57.8±20.1	13.3±9.5	21.6±13.4

Table 6 Accuracy of ST29 and *Kashuf* on the test documents (**RQ3**).

N	Trace Link Level								Requirement Level						
	ST29				<i>Kashuf</i>				ST29			<i>Kashuf</i>			
	T*	TP	FP	R	T*	TP	FP	R	EM	PM	SR	EM	PM	SR	
RD1	73	57	0	1	0.0	57	10	95	17.5	32	1	45.2	19	16	47.9
RD2	64	65	1	3	0.2	65	11	72	16.9	30	0	46.9	29	5	54.7
RD3	61	43	0	15	0.0	43	7	69	16.3	23	4	44.3	13	16	47.5
RD4	92	86	2	1	0.1	86	8	94	9.3	20	0	21.7	14	10	26.1

*T**: Predicted trace links, EM: Exact Match, PM: Partial Match, SR: Success Rate.

Table 7 Accuracy of RICE-based approach on the test datasets (**RQ4**).

N	Trace Link Level					Requirement Level		
	T*	TP	FP	R	EM	PM	SR	
RD1	73	57	47	136	82.5	2	63	89.0
RD2	64	65	50	207	76.9	2	52	84.4
RD3	61	43	39	197	90.7	0	57	93.4
RD4	92	86	74	208	86.0	0	83	90.2

*T**: Predicted trace links, EM: Exact Match, PM: Partial Match, SR: Success Rate.

研究结论

- 在小规模同构数据上微调的 Sentence-Transformer 可显著优于传统 IR-基线；但缺乏跨领域、跨法规的泛化能力。
- 结合精心设计的提示，零样本 LLM 能在复杂法规背景下提供高召回、可解释的链路推荐，显著减轻工程师负担。
- 因此，未来 LRT 研究应聚焦于提示工程+LLM的范式，并探讨融合领域知识与人-机协同以进一步提升精度和可扩展性。

启发：目测要投稿ESE，特点就是比较全面的settings排列组合，细节描述清晰易懂，不难，不是大创新，是完整、规范的实证研究，非常契合ESE这个期刊论文的特点，正是由于上述特点，这个Springer的二区期刊的论文篇幅一般比较长，长在需要把细节和全面的考虑阐述清楚。



学海拾珠漫步知途