LaajMeter：LaaJ 评估框架

Gal Amram, Eitan Farchi, Shmulik Froimovich, Raviv Gal, and Avi Ziv

IBM Research, Israel

Abstract

大型语言模型 (LLMs) 越来越多地被用作自然语言处理任务中的评估者，这种范式被称为 LLM-as-a-Judge (LaaJ)。虽然在一般领域中效果显著，但在特定领域中，LaaJs 面临着重大挑战，这些领域中标注数据稀缺且专家评估成本高昂。在这些情况下，通常会使用尚未针对其应用领域验证的指标进行元评估。因此，很难确定哪些指标能够有效识别 LaaJ 的质量，以及什么样的阈值表明评估者的表现足够好。

在这项工作中，我们引入了 LaaJMeter，这是一种基于模拟的方法框架，旨在对 LaaJs 进行受控的元评估。LaaJMeter 使工程师能够生成代表虚拟模型和评估者的合成数据，从而在现实条件下系统分析评估指标。这有助于从业者针对特定的评估任务验证并优化 LaaJs：他们可以测试他们的指标是否正确区分出更好或更差的 (虚拟)LaaJs，并估算适当的评估者合格阈值。

我们在涉及遗留编程语言的代码翻译任务中展示了 LaaJMeter 的实用性，显示了不同指标在评估者质量敏感度上的差异。我们的结果突显了常见指标的局限性以及选择原则性指标的重要性。LaaJMeter 为在资源匮乏的环境中评估 LaaJs 提供了一种可扩展且可扩展的解决方案，助力确保自然语言处理中的评估可信赖且可重现的更广泛努力。

1 介绍

大型语言模型（LLMs）极大地拓展了自然语言处理（NLP）的能力，使得翻译、摘要、问答和代码生成等多种任务不仅可行，而且具有可扩展性和可访问性。它们在不同领域和多样化语言输入中的泛化能力使得在传统基于规则或监督的方法先前不实用的环境中能够部署 NLP 工具。

随着自然语言处理系统变得更加普遍，对其输出进行可靠且高效的评估变得越来越关键。重要的是，无论是基于人工智能的还是其他的自然语言处理工具的评估，本身就是一个自然语言处理任务。它包括解释文本输出、评估其正确性、相关性或流畅性，以及做出通常依赖于领域特定知识的细致判断。这个评估层对于验证系统行为、指导模型开发以及确保部署安全至关重要 [14] 。

LLM 现在被广泛用作评估者，这种范式被称为 LLM 作为裁判（LaaJ），其中模型负责对其他系统的输出进行评分、排名或评估 [17] 。这种方法已被证明有效，并在研究和生产环境中越来越多地采用 [5] 。LaaJs 主要由于其可扩展性而受到重视：它们可以提供上下文感知的评估，覆盖大规模数据集而无需人工注释。除了独立评估之外，LaaJs 也被集成到 AI 代理架构中，在这些架构中，它们评估其他组件的输出并影响后续决策。在这样的系统中，LaaJ 的判断可能直接影响代理的行动，因此这些判断的可靠性至关重要 [16] 。

然而，日益依赖 LaaJs 引发了一个关键问题：谁来评估评估者？这个挑战被称为元评估，涉及对 LLM 作为评价者所产生的判断进行评估 [5] 。一个强大的元评估框架必须确定这些判断在不同任务和输入中的可靠性、一致性和意义。一个诱人但有缺陷的解决方案是使用另一个 LLM 来执行这一目的。然而，这样做只是将问题提升到一个更高的层次，形成一个没有独立真实依据的递归循环。没有外部的、与模型无关的评估机制，整个过程的有效性就会受到质疑。因此，原则性和可重复的元评估方法必须突破这种依赖循环，避免依赖 LLM 本身。

尝试进行元评估的一种方法是使用文本相似性指标，如 BLEU、ROUGE、METEOR 和 BERTScore，将 LaaJ 的输出与参考判断进行比较 [13] 。虽然这些指标提供了一种方便的自动化方式来评估文本对齐，但它们在捕捉评估任务中细微而依赖于上下文的性质上显得不足。例如，BLEU 和 ROUGE 严重依赖于 n-gram 的重叠，这可能会对有效的转述进行惩罚或对表面的相似性进行奖励。BERTScore 通过利用上下文嵌入对此进行了改进，但它仍然缺乏评估推理质量、事实正确性或特定领域相关性的能力。这些限制在存在多个有效判断或微妙区别至关重要的任务中尤为明显。例如，写代码的指令在编程语言中可以产生多个正确的解决

方案，但这些解决方案可能在文本上毫无相似之处。因此，为了可靠的元评估，与专家人工注释的对比仍然是黄金标准 [14] 。在实践中，与专家排名（例如，通过 Kendall- τ 或 Spearman- ρ ）的一致性和与黄金标准标签的准确性等指标通常用于将 LaaJ 的性能与人工判断进行基准测试。此外，市售的基准存在于许多常见任务中，例如摘要（例如，CNN/DailyMail）、问答（例如，SQuAD）和代码生成（例如，HumanEval），为原则性评估提供了标准化的数据集和注释 [1, 4] 。

虽然现成的基准为评估 LaaJs 在总结、问答和代码生成等通用任务上提供了强大的基础设施，但它们往往不适合这些任务的领域特定变体。例如，用遗留编程语言生成代码、总结法律文件或回答专门科学领域中的问题可能需要标准基准无法涵盖的领域专业知识和背景 [11, 8] 。这些专业化任务通常由于数据稀缺和专家注释访问受限而受到影响，因为合格评估者的稀缺性和成本较高。这种挑战在低预算项目中尤其突出，因为管理领域特定数据集或邀请专家所需的资源不易获得。因此，在这些环境中进行原则性元评估仍然是一个显著的障碍，要求创造性的解决方案来弥补标准化资源的缺乏。

在缺乏标准化基准和专家注释的情况下，在特定领域环境中的元评估通常依赖于创造性、特定任务的策略。一个说明性方法是使用已知或可控质量的输出，例如来自具有不同能力的模型的响应，并验证 LaaJ 为较好的输出赋予更高的得分。然而，由于领域的特异性，通常很少有文献指导哪些指标是有效的或哪些实验设置能为元评估提供有意义的见解 [7] 。即便某个指标似乎与人工判断相关（虽然这种相关性很难验证），确定一个能够表示足够评估质量的阈值仍然固有地困难。在缺乏明确标准的情况下，实践者需要定义临时的标准，这些标准可能在应用之间差异很大并影响可重复性。

为了解决这一挑战，我们提出了 LaaJMeter，这是一个基于模拟的框架，用于对 LaaJs 进行可控的元评估。LaaJMeter 生成代表虚拟模型和虚拟 LaaJs 的合成数据，从而能够在现实条件下系统地分析评估指标。通过模拟不同质量的模型以及具有不同噪声和偏差特征的 LaaJs，我们可以在不依赖人工标注 [10] 的情况下测试指标的敏感性和鲁棒性。

我们通过代码翻译的一个用例展示了 LaaJMeter 的实用性，在该用例中，一个大型语言模型将遗留代码翻译成现代编程语言。在这种情况下，我们模拟虚拟模型和 LaaJ，并评估几个指标：t 检验、Kendall's τ 相关和排序实验。我们的结果揭示了虽然某些指标（例如 t 检验）对 LaaJ 质量的敏感性不够，但其他指标（例如 τ 相关）即使在模型距离估计不精确时也提供了强有力的信号。尽管排序实验有效，但需要仔细解释模型距离以确保结果具有意义。

总之，LaaJMeter 为评估作为评判者的大型语言模型（LLMs）提供了一个有原则且可扩展的框架，尤其是在标注数据稀少或不可靠的情况下。它补充了现有的评估范式，并为指标选择、阈值设定和基于模拟的基准测试提供了实用指南。我们的工作为基于 LLM 的评估研究增添了新的内容，并为规模化、数据高效的模型判断质量评估开辟了新的方向。

2 相关工作

几项先前的研究共同强调了在专家数据有限或不可用的情况下，为 LaaJs 制定有原则的、适应性强的元评估策略的重要性。

LLM-作为-评判者（LaaJ）模型的范式作为人工评估的可扩展替代方案已获得关注，特别是在专家注释稀缺的领域。虽然像 AlpacaEval 和 Chatbot Arena 这样的通用基准已被广泛采用，但它们往往无法捕捉到领域特定任务的细微差别。工作 [12] 通过引入一个数据管道来构建为 LaaJ 框架量身定制的领域特定评估集，从而解决了这一限制。他们的方法结合了人工策划、半监督聚类和分层抽样，以确保在法律、医学和多语言环境等领域的平衡代表性。

在法律领域，工作 [3] 展示了如何在有限的标注数据条件下利用 LLMs 进行术语提取。他们的研究比较了几种模型的提示工程和微调策略，显示零样本和单样本能力可以显著减少对专家标注数据的依赖。

这项工作 [2] 探讨了使用开源 LLMs 进行政治科学中的文本标注，突出了在资源匮乏的环境中微调的有效性。他们的发现表明，适量的标注数据可以获得有竞争力的表现，为特定领域的评估提供了实际途径。

从更广泛的角度来看，工作 [9] 提供了一个关于大语言模型评估方法的系统调查，识别了可重复性和可靠性方面的关键挑战。他们的建议强调需要标准化的协议，特别是在评估设置差异很大的专业领域中。

最后，[15] 的工作提出了一个贝叶斯评估框架，用于评估 LLMs 作为函数逼近器的能力。他们的工作提供了有关先验领域知识如何影响模型行为的洞见，这对于在特定领域上下文中设计评估器尤为重要。

3 问题描述

特定领域的自然语言处理任务，尤其是在低预算项目中，往往由于缺乏高质量的人类标记数据而受到影响。在这种情况下，由于可靠的真实标签有限，评估 LaaJ 的性能变得具有挑战性。

在缺乏全面专家评估的情况下，一种评估 LaaJ 质量的方法是将其应用于已知并具有不同质量的输出。被评估的 LaaJ 应该能够正确区分同一输入的不同输出，通过给更好的输出赋予更高的评分。这种形式的元评估可以基于其对输出进行真实质量排序的能力在不同的 LaaJ 之间进行比较。有关这种评估的示例，请参见 [6] 。然而，即使相对比较是可能的，仍然很难定义一个满意的阈值来决定什么构成可接受的 LaaJ 质量。

另一个常见的情况是，基准测试由单个专家标注。虽然这样的数据可以用于比较 LaaJ，例如，通过测量其相对于专家评分的准确性，但仍然不清楚一个高质量的 LaaJ 应该达到什么水平的协议。这与有多个专家标注的情况形成对比，在这种情况下，可以根据人类评分者之间的标注一致性来校准 LaaJ 的性能。在缺乏这种比较数据的情况下，很难建立可接受的 LaaJ 质量的有意义的阈值。

因此，核心问题在于当元评价数据稀缺时，就很难确定：

- 给定的度量是否适合评估 LaaJ 质量?
- 对于该指标，应该考虑什么样的阈值可以接受?

4 LaaJMeter，一般描述

为了解决上述问题，我们建议创建代表虚拟模型和虚拟 LaaJs 的合成数据。设计该数据是为了模拟模型和 LaaJs 在特定任务上的行为，实现可控的元评估。关键的设计原则是确保生成的数据与实际分布以及在评估的 NLP 任务中观察到的行为紧密相似。

4.1 虚拟点

我们首先定义一组虚拟点，这些点作为虚拟模型的输入。这些点并没有被实例化为实际内容，而是仅仅指定这个集合的大小。一个合理的选择是使其大小与用于任务评估的典型基准相匹配。虚拟点的数量可以影响指标计算的稳定性。例如，较大的集合往往会减少评估指标平均分数的方差。

一旦虚拟点的数量确定，我们定义虚拟模型。从概念上讲，一个虚拟模型为每个虚拟点产生推断，然后分配真实评分。在实践中，我们不实际生成推断，而是直接分配评分。正式地，一个虚拟模型被定义为函数 $M : \{0, \ldots, n-1\} \to [a, b]$ ，其中 n 是虚拟点的数量，$[a, b]$ 是可能的评分范围。一个好的设计选择是确保评分分布反映评估任务中观察到的典型模式。

为了模拟质量不同的模型，我们通过系统地修改现有模型的评分来生成额外的虚拟模型。例如，考虑一个模型 M ，我们可以定义一个新模型 M' ，使得 $M'(k) = M(k) + b_k$ ，其中 b_k 是从一个分布中抽样的。如果该分布趋向于正值，M' 代表质量高于 M 的模型。重要的是，这种方法保持了模型之间的逐点关系，确保 M' 在期望上在所有点上都表现优于 M ，模拟改进版本模型的行为。

4.2 虚拟 LaaJs

在定义了虚拟模型后，我们继续构建虚拟 LaaJs。直观地来看，一个虚拟 LaaJ 评估由虚拟模型生成的推论。形式上，一个虚拟 LaaJ L 是一个映射 $L(i, k) = s$ ，其中 i 是虚拟模型的索引，k 是虚拟点的索引，s 是 LaaJ 对模型在该点的推论所分配的分数。虚拟 LaaJ 的质量由其分数与虚拟模型的真实分数之间的预期偏差决定。通过生成不同质量的 LaaJs，我们可以评估哪些指标在区分它们时是有效的。

5 代码翻译用例

现在，我们描述一种具体实现上述技术的方法，该方法应用于评估设计用于将遗留编程语言翻译为现代编程语言的 LLM。

5.1 虚拟模型

我们首先构建了一个虚拟的基础模型 $M_0 : \{0, \ldots, 99\} \to [0, 30]$ 。该模型通过为点索引分配真实得分来创建，从我们为代码翻译任务开发的 LaaJ 的得分分布中采样。因此，我们选择范围 $[0, 30]$ 来匹配该 LaaJ 使用的评分尺度。100 个点的数量是我们用于模型评估的基准测试中典型的标准。

在构建了 M_0 之后，我们生成了附加的虚拟模型 $M_1, \ldots, M_{20}$ 。每个模型 M_{i+1} 都通过如下方式从 M_i 导出：对于每个点 k ，以概率 p ，我们设定 $M_{i+1}(k) = \min\{M_i(k) + 1, 30\}$ ，在不超过最大值 30 的情况下增加分数。以概率 $1 - p$ ，我们设定 $M_{i+1}(k) = \max\{M_i(k) - 1, 0\}$ ，在不低于 0 的情况下减少分数。选择 p 的值是为了使得 M_{i+1} 的期望平均分比 M_i 高 0.5 分，并考虑到在边界（0 和 30）上的分数数量。

然后，我们构建了比 M_0 得分更低的模型，记为 $M_{-1}, \ldots, M_{-20}$ ，使用相反的程序进行。在每个 M_{i-1} 中，选择 p 的值，使其期望平均得分比 M_i 低 0.5 分。总共，我们创建了 41 个虚拟模型：$M_{-20}, \ldots, M_{20}$ ，质量逐步变化。对于任何一对 M_i 和 M_j ，若 $-20 \leq i < j \leq 20$ ，则 M_j 的平均得分超过 M_i 的平均得分 $\frac{j-i}{2}$ 。我们将值 $j - i$ 称为 M_i 和 M_j 之间的距离。

5.2 LaaJ 模拟

我们现在描述虚拟 LaaJ 的构建，这是基于我们在代码现代化任务中评估 LaaJs 的经验。在实际操作中，基准测试通常包含简单和具有挑战性的例子混合。一个称职的 LaaJ 通常在简单情况下表现良好，而在涉及特定功能的例子中，如罕见的命令，这种功能增加了评估难度，其表现可能有所不同。有些 LaaJs 可能能够有效处理某些功能而在其他方面有困难。

为了模拟这种行为，我们将 100 个虚拟点中的 20 个指定为“简单点”，所有虚拟 LaaJ 都应准确评估这些点，而不考虑它们的质量。剩下的 80 个点被分成 10 个互不相交的集合，每个集合包含 8 个点，被称为特征集合。然后我们构建了虚拟 LaaJ $L_1, \ldots, L_{10}$ ，如下所示：对于每个 LaaJ L_j ，我们随机选择 j 个特征集，在这些集上它表现较差。在这些集上，L_j 的评分表现出偏差和高噪声；在所有其他点上，它只表现出低噪声。

形式上，对于不在 L_j 所选特征集合中的点 k ，得分计算为 $L_j(i, k) = M_i(k) + l_{i,k}$ ，其中 $l_{i,k} \sim \mathcal{N}(0, 1)$ 表示低噪声。回想一下，$M_i(k)$ 是模型 M_i 在点 k 上的真实得分。

如果 k 属于为 L_j 选择的特定集之一，我们为整个集 S 固定采样一个偏差 $b_S \sim \mathcal{N}(0, 2)$ ，并将分数计算为 $L_j(i, k) = M_i(k) + b_S + h_{i,k}$ ，其中 $h_{i,k} \sim \mathcal{N}(0, 5)$ 表示高噪声。此设置使我们能够模拟对特定特征敏感性不同的 LaaJs，从而实现对度量效果的稳健评估。注意，对于 $i < j$ ，L_i 优于 L_j ，因为 L_j 在额外的 $j - i$ 特征集中引入了偏差和高噪声，相应地比 L_i 多 $8 \times (j - i)$ 个点。

我们为用于 LaaJs 元评价的各种指标提供了模拟。

5.3 t -测试

我们从一个消极结果开始，这一结果展示了某个特定度量用于元评估的局限性。具体来说，我们使用 t 检验来评估 LaaJ 的质量，该检验应用于不同质量的模型推理的 LaaJ 评分。

思路是这样的：我们选择两个模型（或模型版本），其中一个已知比另一个好。我们应用一个 LaaJ 来评估两个模型的推断，并计算得到的得分向量的 p 值。期望是一个高质量的 LaaJ 会产生一个统计上显著的结果，表明较好的模型的得分更高。然而，一个不好的 LaaJ 可能无法检测到这种差异。

重要的是，模型之间的距离也起着关键作用。如果距离过大，即使是较差的 LaaJ 也可能检测出差异，从而导致显著的 p 值。为了探讨这一点，我们模拟了在虚拟 LaaJ 分数上跨不同距离的虚拟模型对进行的 t 检验。结果的 p 值如表 1 所示。

5.3.1 t 测试结果分析与结论

如表 1 所示，所有的 LaaJ 在模型距离大于 2 时均提供统计上显著的 p 值。这表明即使是像 L_{10} 这样较差的 LaaJ，也能检测出模型之间的差异，即使这种差异仅仅是显著而已。因此，t 测试对 LaaJ 的质量缺乏敏感性，无法区分性能不同的 LaaJ。我们得出结论，t 测试在此背景下不适合进行元评价。

Distance/LaaJ	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
1	0.07	0.08	0.11	0.19	0.20	0.20	0.21	0.25	0.24	0.29
2	0.00	0.00	0.02	0.04	0.04	0.06	0.05	0.09	0.09	0.13
3	0.00	0.00	0.00	0.01	0.01	0.01	0.02	0.01	0.02	0.05
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 1: p 值结果

5.4 Kendall-τ 等级相关

我们现在展示一个正面的例子，其中我们的模拟支持一个度量的有用性。Kendall-τ 等级相关系数（也称为 τ -相关性）衡量两个排序向量之间的一致性，其值范围从 -1 （完全不一致）到 1 （完美一致）。

对于元评估，我们使用 τ -相关性，具体如下：我们选择两个模型（或模型版本）并将它们应用于相同的基准。接着，我们应用一个 LaaJ 来评估推理，并计算结果分数向量之间的 τ -相关性。由于推理质量也依赖于输入的复杂性，我们期望分数的排序在某种程度上会被保留。高质量的 LaaJ 应产生能够保持这种排序的分数，从而导致较高的 τ -相关性。相比之下，劣质的 LaaJ 可能无法检测到低质量模型输出中的问题，从而导致较低的相关性。

此外，我们预期在比较小距离的模型时，良好和较差的 LaaJ 之间在 τ 相关性上的差异会更加明显。在这种情况下，推理质量的细微差异更难以检测，而只有更好的 LaaJ 才能够有效地捕捉到这些差异。

我们的模拟结果展示在表 2 中。

Distance/LaaJ	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
1	0.79	0.76	0.71	0.67	0.65	0.60	0.57	0.56	0.53	0.48
2	0.78	0.74	0.70	0.67	0.63	0.60	0.58	0.55	0.53	0.48
3	0.76	0.73	0.69	0.65	0.63	0.59	0.56	0.56	0.52	0.47
4	0.75	0.72	0.67	0.64	0.61	0.58	0.55	0.55	0.52	0.48
5	0.74	0.71	0.67	0.63	0.62	0.57	0.55	0.55	0.52	0.46
6	0.73	0.70	0.66	0.63	0.61	0.57	0.54	0.53	0.52	0.46
7	0.72	0.69	0.65	0.62	0.61	0.56	0.53	0.53	0.50	0.46
8	0.70	0.68	0.65	0.60	0.60	0.56	0.53	0.53	0.51	0.46
9	0.69	0.67	0.63	0.60	0.59	0.55	0.53	0.53	0.50	0.44
10	0.68	0.67	0.62	0.59	0.58	0.54	0.52	0.50	0.49	0.45

Table 2: Kendall-τ 等级相关系数结果

5.4.1 τ -相关结果分析与结论

结果符合我们的预期。更好的 LaaJ 会持续产生更高的 τ 相关性分数，特别是在模型距离较小的情况下。值得注意的是，该度量对 LaaJ 质量的敏感性大于对模型距离的敏感性，这使得它在元评估中特别有用。这种敏感性允许从业者在没有模型距离的精确知识的情况下有效地使用 τ 相关性。

我们的研究结果表明，即使模型距离只是粗略估计，一个约为 τ 的 0.70 -相关分数也足以区分高质量和低质量的 LaaJs。这一见解支持将 τ -相关性作为可靠指标的使用，并有助于建立 LaaJ 评估的实际阈值。我们得出的结论是，τ -相关性是 LaaJ 元评估的有效和稳健的指标。

5.5 排序实验

在我们最后的例子中，我们展示了对距离的敏感性，从而学会适当地应用给定的度量。我们用于元评估的测试是 LaaJ 能否正确识别两个候选模型中更佳的模型。为此，我们选择两个已知且质量相近的模型，将它们应用于相同的基准，并计算出 LaaJ（弱地）更倾向于较佳模型的数据点的百分比。

值得注意的是，更好的模型不一定在所有点上都能产生更优的推理。然而，我们期望更准确的 LaaJ 在此测试中能够产生更高的结果，因为它噪声更小，错误更少，因此更有可能倾向于更好的模型。我们使用虚拟模型在不同距离进行了排序实验的模拟。结果如表 3 所示。

Distance/LaaJ	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
1	64.8 %	64.6 %	62.5 %	62.4 %	61.6 %	61.1 %	59.6 %	59.5 %	58.8 %	57.2 %
2	74.3 %	71.5 %	70.6 %	68.4 %	67.1 %	67.3 %	65.3 %	64.5 %	63.4 %	61.1 %
3	78.2 %	77.0 %	77.2 %	74.7 %	71.8 %	70.9 %	69.9 %	68.3 %	67.5 %	63.9 %
4	83.2 %	80.4 %	79.7 %	77.5 %	76.3 %	74.1 %	73.1 %	72.0 %	70.8 %	67.5 %
5	86.2 %	84.6 %	82.5 %	81.1 %	78.9 %	76.7 %	75.8 %	73.9 %	72.8 %	69.8 %
6	88.4 %	86.9 %	85.5 %	83.5 %	82.3 %	79.4 %	78.5 %	76.9 %	75.7 %	71.8 %
7	90.3 %	88.9 %	86.9 %	86.0 %	83.9 %	81.7 %	79.9 %	78.6 %	77.1 %	74.4 %
8	91.6 %	90.2 %	89.2 %	86.8 %	86.3 %	83.6 %	82.3 %	81.5 %	81.1 %	76.0 %
9	92.9 %	91.3 %	90.6 %	88.4 %	87.1 %	85.1 %	84.5 %	82.0 %	82.2 %	77.6 %
10	94.4 %	92.7 %	92.2 %	89.8 %	88.4 %	86.9 %	85.5 %	84.2 %	83.7 %	79.9 %

Table 3: 实验结果排序

5.5.1 排序实验结果分析与结论

我们的结果表明，排序实验对 LaaJ 质量十分敏感，因此是一个对 LaaJ 元评估有用的指标。然而，该指标对模型距离也很敏感，这使得结果的解释变得复杂。需要仔细估计模型距离以设定一个有意义的评估阈值。

例如，在距离 4 时，L_3 达到大约为 80% 的结果，而在距离 10 时，更嘈杂的 L_{10} 也能达到该结果。这说明如果不考虑模型距离，度量可能无法区分不同质量的 LaaJ。

总之，排序实验是一个有效的衡量标准，但其应用需要谨慎处理。特别是，对模型之间相对距离的理解是必要的。

6 结论

在这项工作中，我们解决了在高质量标注数据稀缺的特定领域 NLP 任务中评估大型语言模型作为评判者（LaaJs）的挑战。我们引入了 LaaJMeter 框架，通过构建代表虚拟模型和 LaaJs 的合成数据，实现了可控的元评估。这种方法允许在实际条件下对评估指标进行系统化分析。

通过一个详细的代码翻译用例，我们展示了如何使用 LaaJMeter 来模拟细致入微的模型和 LaaJ 行为。我们评估了几个指标，t -测试、Kendall- τ 相关性以及一个排序实验，突出了它们的优点和局限性。我们的研究结果表明，尽管 t -测试对 LaaJ 质量缺乏敏感性，但 Kendall- τ 相关性即使在模型距离估计不精确的情况下也能提供有力的信号。排序实验虽然有效，但需要根据模型距离进行仔细校准。

在本文中，我们专注于模拟不依赖于人工标注数据的评估指标。然而，我们的框架可以扩展以结合此类数据。此扩展将涉及模拟专家评分。一种简单的方法是定义地面真相评分为人工评价。或者，可以在地面真相评分中加入噪声以模拟专家判断的变异性。这使得可以评估人工参考指标，例如均方误差。我们将探索此方向留作未来工作。

References

[1] Confident AI. Squad | deepeval - the open-source llm evaluation framework, 2025. Accessed August 2025.

[2] Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Mohammadmasiha Zahedivafa, Juan D. Bermeo, Maria Korobeynikova, and Fabrizio Gilardi. Open-source llms for text annotation: A practical guide for model setting and fine-tuning. *Journal of Computational Social Science*, 8(17), 2025.

[3] Julien Breton, Mokhtar Mokhtar Billami, Max Chevalier, Ha Thanh Nguyen, Ken Satoh, Cassia Trojahn, and May Myo Zin. Leveraging llms for legal terms extraction with limited annotated data. *Artificial Intelligence and Law*, 2025.

[4] Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[5] Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *arXiv preprint arXiv:2401.16788*, 2024.

[6] Ora Nova Fandina, Eitan Farchi, Shmulik Froimovich, Rami Katan, Alice Podolsky, Orna Raz, and Avi Ziv. Automated validation of llm-based evaluators for software engineering artifacts, 2025.

[7] Rishav Hada, Varun Gumma, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. METAL: towards multilingual meta-evaluation. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2280–2298. Association for Computational Linguistics, 2024.

[8] Swayambhoo Jain, Ravi Raju, Bo Li, Jonathan Li, and Urmish Thakker. Constructing domain-specific evaluation sets for llm-as-a-judge. *arXiv preprint arXiv:2408.08808*, 2024.

[9] Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, Enamul Hoque, Shafiq Joty, and Jimmy Huang. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 13785–13816. Association for Computational Linguistics, 2024.

[10] Yuxuan Lu, Jing Huang, Yan Han, Bingsheng Yao, Sisong Bei, Jiri Gesi, Yaochen Xie, Zheshen Wang, Qi He, and Dakuo Wang. Prompting is not all you need! evaluating llm agent simulation methodologies with real-world online customer behavior data. *arXiv preprint arXiv:2503.20749*, 2025.

[11] Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3016–3054. Association for Computational Linguistics, 2023.

[12] Ravi Raju, Swayambhoo Jain, Bo Li, Jonathan Li, and Urmish Thakker. Constructing domain-specific evaluation sets for llm-as-a-judge. In *Proceedings of the 1st Workshop on Customizable NLP (CustomNLP4U)*, pages 167–181. Association for Computational Linguistics, 2024.

[13] Hadeel Saadany and Constantin Orasan. Bleu, meteor, bertscore: Evaluation of metrics performance in assessing critical translation errors in sentiment-oriented text. In *Proceedings of TRITON (Translation and Interpreting Technology Online)*, pages 48–56, 2021.

[14] Hendrik Schuff, Lindsey Vanderlyn, Heike Adel, and Ngoc Thang Vu. How to do human evaluation: A brief introduction to user studies in nlp. *Natural Language Engineering*, 29(5):1199–1222, 2023.

[15] Shoaib Ahmed Siddiqui, Yanzhi Chen, Juyeon Heo, Menglin Xia, and Adrian Weller. On evaluating llms' capabilities as functional approximators: A bayesian evaluation framework. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pages 5826–5835. Association for Computational Linguistics, 2025.

[16] Huilin Wang and Lei Yu. Can we employ llm to meta-evaluate llm-based evaluators? a preliminary study. In De-Shuang Huang, Bo Li, Haiming Chen, and Chuanlei Zhang, editors, *Advanced Intelligent Computing Technology and Applications*, pages 161–172, Singapore, 2025. Springer Nature Singapore.

[17] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.