arXiv:2601.00223v1 [cs.CL] 1 Jan 2026

JP-TL-Bench：用于日英双向翻译的锚定式大语言模型成对评估

Leonard Lin、Adam
Lensenmayer，Shisa.AI

摘要

我们推出**JP-TL-Bench**，这是一个轻量、开放的基准测试，旨在指导**迭代开发**日语↔英语翻译系统。在这一背景下，常见的挑战是“这两个不错的译文哪个更好？”，而非“这个译文是否合格？”。这一区别对日语↔英语翻译尤为重要，因为礼貌表达、会话含义、省略和语域上的细微选择，都会显著影响人们感知到的自然度。JP-TL-Bench采用了一套协议，旨在让大语言模型评判既**可靠**又**经济实惠**：它通过**无参考译文的成对大语言模型比较**，将候选模型与**固定的版本化锚点集**进行评估。成对比较结果通过**Bradley–Terry模型** [1] 聚合，并以胜率和归一化的**0–10“LT”分数**呈现；该分数由拟合所得对数强度经逻辑变换得出。由于每个候选模型都依据同一冻结锚点集进行评分，只要基础集、评判者和聚合代码相同，评分就会**在结构上保持稳定**。

1. 引言

在开发 Shisa V2 日英双语模型时， [2, 3, 4], 我们发现，现有评估方法无法回答一个实际问题：“这两种质量都不错的译文，哪一种更好？”大语言模型（LLM）大幅提升了机器翻译（MT）的质量，但许多机器翻译评估方法却未能跟上这一进展。llm-jp-eval[5] 等评测套件提供了有价值的基于COMET的验证，但这些基准测试在很大程度上已趋于饱和：分数十分接近，往往无法区分高质量译文。

机器翻译领域高度依赖 BLEU [6], 、chrF [7], 等基于参考译文的指标，以及 COMET [8]等学习型指标。这些指标有助于广泛验证，但近期研究表明，它们可能无法准确反映顶尖译文的质量，也并非为已接近流利水平的输出提供高分辨率信号而设计 [9, 10, 11]。现代前沿大语言模型能够识别细微差别，因此很适合作为无参考译文的评审者——然而，若不谨慎设计协议，大语言模型裁判 [12] 可能并不可靠 [13]。

1.1 概述

我们推出 **JP-TL-Bench**，这是一套基准测试和协议，专为**迭代改进日语↔英语翻译质量**这一特定用途设计，适用于绝对指标难以解读的情况。大语言模型裁判进行成对比较，具有很强的区分度和可靠性；这里的关键思路是与冻结锚点集进行比较，而非进行所有两两比较，从而以固定的 O(N) 成本得出稳定的绝对分数。Chatbot Arena [14] 等大规模偏好排行榜已证明成对比较的实用性，但其基于Elo的排名受比较顺序影响，会产生

随候选池变化而漂移的浮动分数，且成本规模为 $O(N^2)$。通过使用固定锚点集，JP-TL-Bench 避免了这三个问题。

该基准测试包含 70 个翻译条目，涵盖 **英语→日语和日语→英语** 以及 **简单/困难**难度层级。每次评测运行对每个候选模型大约需要 $\mathbf{70 \times 20 = 1{,}400}$ 次成对判断（与评估模型的数量呈线性关系），因此该基准测试适用于迭代微调。我们还介绍了实践中常用的两个互补基线——**llm-jp-eval** [5]（以 COMET 为核心的套件）和 **基于评分标准的大语言模型评测**（用于我们的 LiquidAI 黑客松工作流）——并在附录中提供提示词/评分标准模板，将其与基于评分标准训练的评判者以及 Prometheus [15]和 UltraFeedback [16]等反馈数据集联系起来。

1.2 贡献

- **日语**到↔英语翻译评测的锚定式成对比较协议：候选模型不是与浮动池进行比较，而是与固定的版本化锚点集进行比较。
- **无参考译文评测**采用透明的提示词和确定性解码；该基准测试不使用参考译文。
- **通过 Bradley–Terry 进行分数聚合和报告** [1]，并采用归一化 0–10 LT 分数，以确保在固定条件下的可解释性和可比性。
- **精心筛选的锚定集合**：基础集 v1.0 从数百个模型中选出，旨在覆盖广泛的质量范围，并使胜率分布均匀（截至 2025 年年中）。
- **版本化比较池**：锚点集采用语义化版本控制，以便进行修复和改进，同时保持不同评测之间的可比性。
- **开源实现**：JP-TL-Bench 已用于开发多个 Shisa.AI 模型版本，并以 Apache 2.0 许可证开放获取，地址为 `https://github.com/shisa-ai/jp-tl-bench`.
- **实用集成指南**：介绍 JP-TL-Bench 如何与 llm-jp-eval [5] 和基于评分标准的大语言模型评测互为补充（附录 C）。

2. 背景与相关工作

2.1 基于参考译文的机器翻译指标及其有效性问题

尽管已知存在局限性（表层重叠、对分词和有效释义的敏感性），BLEU [6] 和 chrF [7] 仍被广泛使用。事实上，WMT22 指标共享任务得出的结论是，重叠指标与人工评分的相关性较差，并建议不再局限于这些指标 [17]。在 WMT 风格设置下，COMET [8]等学习型指标与人工判断的相关性较好，但近期分析对这些指标在高质量区间的表现以及评测陷阱提出了担忧 [9, 10]。更广泛地说，BLEU 作为评测工具的有效性已受到严格审视 [11]。

COMET 系列还包括无参考质量估计变体，例如 COMETKiwi[18]，以及更透明的变体，例如 xCOMET [19]；在第5.2节中，我们将 COMET 评分与 JP-TL-Bench 的大语言模型评审结果进行比较。

2.2 人工评估框架

人工评估仍是黄金标准。MQM 为细粒度错误标注提供了结构化框架 [20], ，而直接评估技术则建立了连续的人工评分规程[21]。然而，专家评估成本高昂且难以扩展；大规模证据凸显了获得可靠人工判断的复杂度 [22]。

2.3 基于大语言模型的机器翻译评估

基于大语言模型的评估已成为一种可扩展的替代方案。Kocmi 和 Federmann 表明，强大的大语言模型可以充当高性能翻译评估器 [23], ，而 GEMBA-MQM 则通过 MQM 风格的错误片段扩展了这一思路 [24]。BatchGEMBA 探索了通过批处理和提示压缩实现令牌高效评判的方法 [25]。

2.4 大语言模型裁判的可靠性、偏差与评分标准

大语言模型裁判已在一般评估场景中得到研究 [12, 26] ，也有相关综述 [13]。研究记录了裁判偏差和不公平问题 [27], ，而位置偏差是比较评估场景中一种已知的失效模式 [28]。另一条相关研究路线是利用评分标准和反馈数据集训练或蒸馏专用裁判，例如 Prometheus [15]和 UltraFeedback [16]。

2.5 成对偏好排行榜与排序模型

Chatbot Arena [14]让大规模成对偏好评估广为流行。偏好聚合通常采用 Bradley–Terry式模型 [1] 或埃洛评分式更新 [29]。翻译基准测试的一个关键区别在于**分数稳定性**：随着模型池变化，浮动池排行榜可能会发生漂移，而锚定基准则以牺牲部分灵活性为代价，换取可比性。

FiRE [30]提出了面向机器翻译的细粒度无参考排序评估，在方法上与之十分接近；JP-TL-Bench 则通过明确将分数锚定到冻结的基础集快照，来支持稳定迭代。

2.6 日语机器翻译套件与基准

FLORES/NLLB 提供了包括日语在内的多语言机器翻译基准 [31, 32]。llm-jp- eval 项目 [5] 提供了日语评估套件，其中包括机器翻译任务和标准化评分（以 COMET 为核心），并常被用作机器翻译验证的参考实现。JParaCrawl [33] 等平行数据资源有助于训练和评估，但自身无法解决细粒度区分问题。

日语↔英语翻译面临着句子级评估往往无法捕捉的挑战。日语是一种代词省略语言，主语和宾语经常被省略（即零代词），需要根据话语语境进行推断 [34]。日→英系统经常无法正确处理这些情况，产生性别误判代词或语义角色互换等错误，而 BLEU 等句子级指标无法捕捉这些错误 [35]。日语对语域也高度敏感：敬语通过动词形态表达尊敬、正式程度和社会距离。由于英语缺乏语法化敬语，英→日系统经常产生不恰当的正式程度——输出过于随意，或错误地使用敬语——而且没有显式控制机制 [36]。

2.7 总结比较

表1： 机器翻译评估方法

方法类别	方法	评判者	Cost	局限性
BLEU/chrF	与参考译文的 N元语法重叠	算法式	Low	会惩罚有效释义
COMET	神经嵌入相似度	神经网络模型	Low	在高质量水平上趋于饱和
MQM人工评估	专家错误标注	人工	High	成本高、速度慢
GEMBA	大语言模型绝对评分 (0-100)	LLM	中等	高分段的分数压缩
Chatbot Arena	成对比较 + 浮动埃洛评分	人工	High	依赖顺序，分数漂移，$O(N^2)$
JP-TL-Bench	成对比较 + 固定锚点	LLM	Low	依赖基础集/评判者

3. JP-TL-Bench基准测试设计

3.1 任务与条目

JP-TL-Bench 包含 70 个翻译条目，旨在重点考察日语特有现象（语域/敬语、歧义消解、文化适配、技术术语）。条目分为：

- **方向**：EN→JA（34 个条目）和 JA→EN（36 个条目）
- **难度**：简单（30 个条目）与困难（40 个条目）

按类别划分：EN→JA 简单（15），EN→JA 困难（19），JA→EN 简单（15），JA→EN 困难（21）。

作者在审阅现有机器翻译语料库并观察 Shisa 模型此前部署时在实际应用中出现的失效模式后，构建了该测试语料库。条目选择由一位作者指导，他拥有超过 10 年的日语专业翻译经验，重点关注能够区分合格翻译与优秀翻译的语言现象。示例条目见附录 A。

条目集特意设计得**足够小，**以便在模型迭代期间频繁运行，同时也**具有足够的针对性，**能够区分表现强劲的系统，而语料库级指标往往难以充分拉开差距。

3.2 锚定集合（Base Set v1.0）

该基准测试的核心稳定机制是一组由 20 个模型组成的冻结**锚定集合**。这组模型包含强弱系统，以提供宽广的动态范围，并避免在高端过度聚集。

构建过程：Base Set v1.0 源自最初的 v0.9 快照，该快照汇集了近 200 个模型的翻译输出和两两比较结果。我们从中手动挑选了 20 个锚点，使胜率间距大致均匀（间隔为 ~5%，范围从 ~2% 到~96%），同时兼顾 EN→JA 和 JA→EN 两个方向的表现。这有时颇具挑战，因为方向性表现可能高度不对称（例如，某个模型擅长 JA→EN，但不擅长 EN→JA）。最终的选择优先考虑在两个方向上表现较为均衡，或在各个方向上代表不同质量层级的模型。

v1.0 清单、翻译和评分报告可在代码仓库中找到，地址为 https://github.com/shisa-ai/jp-tl-bench 位于 `baseset/v1.0/`。以下锚点胜率和 LT 分数取自已发布的 v1.0 报告，对应于**gemini-2.5-flash** 评判者，其**温度为 0**。

表 2： Base Set v1.0 锚点（总体切片）

#	模型	胜率	LT
1	google/gemini-2.5-pro	96.15%	9.94
2	google/gemini-2.5-flash	92.93%	9.89
3	Qwen/Qwen3-30B-A3B-Instruct-2507	84.37%	9.63
4	shisa-ai/shisa-v2-llama3.1-405b	81.46%	9.49
5	openai/gpt-4o	76.04%	9.12
6	shisa-ai/shisa-v2-unphi4-14b	72.82%	8.83
7	tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.5	62.14%	7.42
8	nvidia/NVIDIA-Nemotron-Nano-12B-v2	59.94%	7.05
9	meta-llama/Llama-3.3-70B-Instruct	58.05%	6.72
10	microsoft/phi-4	49.80%	5.12
11	cyberagent/Mistral-Nemo-Japanese-Instruct-2408	47.60%	4.67
12	Qwen/Qwen3-4B	44.78%	4.10
13	LiquidAI/LFM2-2.6B	43.83%	3.91
14	meta-llama/Llama-3.1-8B-Instruct	38.81%	2.94
15	microsoft/Phi-4-mini-instruct	24.98%	0.99
16	augmxnt/shisa-7b-v1	21.44%	0.68
17	meta-llama/Llama-3.2-3B-Instruct	19.24%	0.54
18	Rakuten/RakutenAI-2.0-mini-instruct	14.23%	0.29
19	LiquidAI/LFM2-350M	8.88%	0.14
20	SakanaAI/TinySwallow-1.5B	2.51%	0.04

版本管理：锚点集遵循语义化版本控制。补丁版本（例如 v1.0.1）可以修正数据错误或回填缺失的评审结果，而不更改锚点模型集；分数仍可比较。次版本（例如 v1.1）可以调整锚点构成，同时保持大致的校准状态。主版本（例如 v2.0）表示启用了新的锚点池，其分数无法与先前版本直接比较。这一版本管理约定使研究人员能够引用特定快照，同时也让基准测试得以不断演进。

4. 评估协议

4.1 配对构建与 A/B 随机化

对于每个条目，都会将候选模型输出与每个锚点模型输出进行比较，为每个候选项生成~1,400 个配对（70×20）。为减轻位置偏差，比较器**使用固定随机种子，每个配对随机决定哪一侧为“译文 A”、哪一侧为“译文 B”**（种子=42），以确保可复现性；无需按两种顺序对每个配对进行双重评审。

4.2 大语言模型评审提示词与解码

JP-TL-Bench 使用固定的比较提示词（附录 B），重点评估八个维度（准确性、自然度、语气／语域等）。此前的版本采用了受 PoLL 启发的本地大语言模型评审团方法

[37], 但从 Base Set v1.0 起，默认评判者为 `gemini-2.5-flash`，并采用确定性解码（温度为 0）。确定性解码可以减少运行间方差，使稳定性主要取决于比较集和聚合。

评估成本与时间：在 ~1 的条件下，每个候选模型进行 400 次评判，且 gemini-2.5-flash 的定价为每 100 万个输入令牌 0.30 美元、每 100 万个输出令牌 2.50 美元（截至 2025 年末），一次完整评估的成本约为每个候选项 **~7.00 美元**。评估通常需要 **10–30 分钟**，具体取决于模型推理栈、输出长度和评判并发度。详细的成本明细请参见第 4.6 节。

4.3 Bradley–Terry聚合

使用 Bradley–Terry模型 [1] 聚合成对比较结果。令 θ_i 表示模型 i 拟合得到的对数强度。则：

$$P(i \succ j) = \frac{\exp(\theta_i)}{\exp(\theta_i) + \exp(\theta_j)}.$$

该实现通过 `choix` [38] 使用最大似然估计。值得注意的是，**每个候选项均独立评分**：要评估某个候选项在给定基础集版本下的得分，我们会复用该基础集冻结的锚点间评判结果，并加入候选项与锚点间评判结果，然后在此合并图上拟合 Bradley–Terry模型。我们绝不会在单次拟合中混合不同候选项的评判结果，因此添加或移除其他候选项不会改变现有候选项的得分。

4.4 报告：胜率和 LT分数

JP-TL-Bench 报告：

- **胜率**：该切片中的实际获胜次数 / 比赛场次（总体、EN→JA、JA→EN、简单、困难）。
- **LT分数（0–10）**：对中心化对数强度进行逻辑变换：

$$\mathrm{LT}_i = 10 \cdot \sigma(\theta_i - \bar{\theta}), \quad \sigma(x) = \frac{1}{1 + e^{-x}}.$$

这里的 θ 是一个中心化常数；在我们的实现中，我们将其设为该拟合切片中所有模型的拟合对数强度均值（包括 20 个锚点和候选项）。该变换会压缩极端强度，同时保留排序，并提升在有界 0–10 刻度上的可解释性。

关于聚合的说明：总体 LT 和胜率都直接基于所有比赛计算。然而，Bradley-Terry 模型会考虑对手实力——战胜强锚点对拟合分数的贡献大于战胜弱锚点，而胜率则平等对待所有胜利。这可能导致 LT排名与胜率排名之间存在轻微差异。

空输出和评判器拒绝的处理：如果候选模型拒绝翻译或输出为空，裁判提示词会指示其将此视为该候选模型落败（见附录 B）。但是，如果评判者本身拒绝评估某一对候选项（例如，由于源文本触发了安全过滤器），则会将该评判结果排除在聚合之外，而不是惩罚任何一方；这可能会略微影响每个模型的对局数。

4.5 结构稳定性与可比性约定

Elo 式系统 [29] 会出现**分数漂移**：每个模型都有一个初始评级，并根据比赛结果和对手实力进行调整。因此，随着新模型加入浮动池并改变评级分布，同一分数的含义也会随时间变化——今天评级为 1200 的模型，可能无法与六个月前评级为 1200 的模型相提并论。由于 JP-TL-Bench 会将每个候选项与固定锚点集进行比较，而非浮动池，因此在以下条件下，分数在结构上保持稳定（与顺序无关）：

1. **Base Set版本** （例如， `baseset/v1.0`）
2. **裁判模型** ＋ **提示词** ＋ **解码设置** （包括温度）
3. **聚合实现/版本**

固定锚点集是关键所在：它结合了成对偏好判断的区分能力 [12] 与静态基准测试的时间可比性——这是浮动池排行榜无法提供的。

4.6 复杂度与成本

每个候选项需要 70 个提示词 $\times$ 20 锚点 $=$ **1、400 次成对判断**，其规模为 O(N)，而不是完全循环赛所需的 O(N^2)。为估算评审成本，我们使用实际的 v1.0 评审基础集（14,002 次 A/B 判断）中的令牌进行统计，采用完全相同的比较提示词（输入：提示模板 ＋格式化译文）和评审响应（输出：分析）。在这些数据中，每次判断的平均令牌用量为 2,626 个输入令牌和 1,567 个输出令牌。

Using gemini-2.5-flash 的定价（输入每百万令牌 0.30 美元，输出每百万令牌 2.50 美元）对应于：

组成部分	令牌	Cost
输入（2,626 $\times$ 1,400）	3.68M	~1.10 美元
输出（1，567 $\times$ 1，400）	219 万	~5.48 美元
每个模型的总计	587 万	~**$6.59**

由于模型始终与规模相同且数量有限的比较池进行比较，每个候选项的评判成本基本保持不变，因此在模型开发期间运行 JP-TL-Bench 是可行的。

4.7 评判质量

在开发过程中，我们将大语言模型裁判的结果与部分条目的母语者双语评估进行了比较，发现两者在评级和评估者间一致性方面都相当。

尽管如此，自动化基准测试使我们能够以合理的成本，对训练过程中生成的数百个消融实验进行评估；为此，我们为 JP-TL-Bench 构建了多个终端界面（TUI）工具，以便对评判结果和模型输出进行便捷的检查与比较。数值分数或胜负记录无法替代对实际模型输出的审视，我们鼓励所有研究人员这样做（也鼓励所有基准测试创建者构建便于定性检查的工具）。

5. 快照分析与基准评判

5.1 锚定集合中的动态范围

设计 Base Set v1.0 锚定集合时，一个关键考量是使其同时具备最大的“动态范围”（LT ≈ 0.04 至 9.94），并让胜负比的间距相对均匀，从而在整个质量范围内实现更均衡的区分度。这一点很重要，因为集合选择不当会导致分数聚类，难以提供有用的信息。

表 3： Base Set v1.0 的示例切片分数（LT）

模型	EN→JA	EN→JA 简单	英语→日语 难	日语→英语	日语→英语 简单	日语→英语 难
gemini-2.5-pro	9.97	9.95	9.99	9.89	9.79	9.99
Llama-3.1-Swallow-8B-Instruct-v0.5	8.80	8.68	9.07	5.96	5.54	6.33
LFM2-2.6B	5.22	6.38	4.06	2.97	3.86	2.07
Llama-3.1-8B-Instruct	1.40	1.07	1.68	4.52	5.87	3.58

出现了两种反复出现的模式：

- **方向不对称性**：一些模型在一个方向上的表现明显更强。Llama-3.1-8B-Instruct 在 JA→EN 上的得分为 4.52，而在 EN→JA 上仅为 1.40，差距达 3.1 分。Llama-3.1-Swallow-8B 则呈现相反的模式：EN→JA 为 8.80，而 JA→EN 为 5.96（差距为 2.8 分），这表明它针对英译日进行了更大程度的优化。

- **等级敏感性**：困难提示往往会扩大较弱模型之间的差距，并暴露出在简单提示上无法观察到的脆弱性。LFM2-2.6B 在 EN→JA 简单提示上的得分为 6.38，但在 EN→JA 困难提示上降至 4.06——这是一种在小型模型中很常见的模式，并且随着模型能力增强而变得不那么明显（对比 gemini-2.5-pro 在简单提示/困难提示下 9.95/9.99 的一致性）。

这些模式为特定部署场景下的模型选择提供依据，并揭示汇总指标无法反映的质量维度。

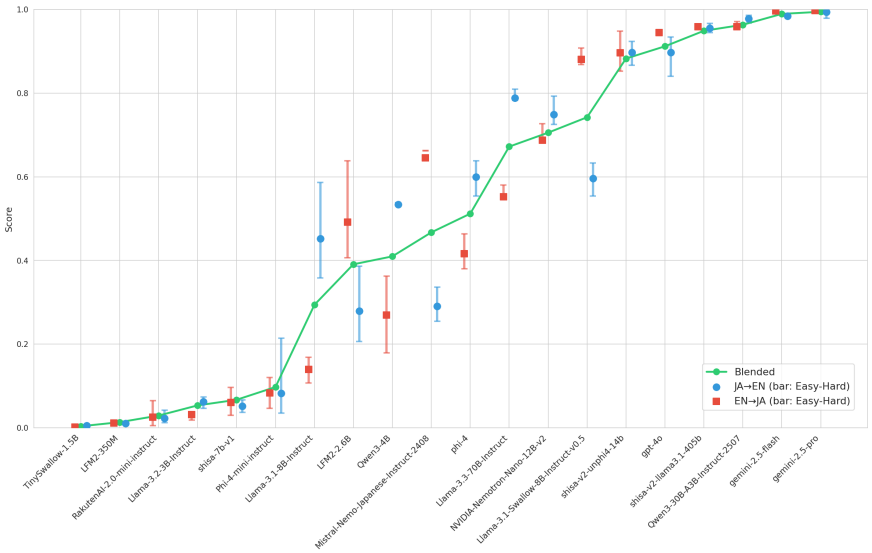

图 1： 按翻译方向和难度集划分的 JP-TL-Bench 分数。按翻译方向区分的 JP-TL-Bench LT 分数（JA→EN 为蓝色，EN→JA 为红色），烛台条形图表示从简单到困难的范围。混合总分以绿色显示。

5.2 与 COMET 指标的比较

JP-TL-Bench 的运行仍然需要更多资源，并不旨在取代 COMET 等学习型指标 [8] 或 llm-jp-eval 等评测套件 [5]。这些工具发挥着互补作用：

- **使用 COMET/llm-jp-eval** 在大型语料库上进行广泛验证和回归检测。
- **当候选项根据自动指标已具备足够性能时，使用 JP-TL-Bench** 进行高分辨率迭代。

话虽如此，COMET 系列指标存在显著局限性，既体现在对顶尖性能的区分能力方面，也体现在压缩的报告范围方面。为展示这一点，我们在基础集 v1.0 翻译上运行了基于 COMET 的评测，以便与我们的 JP-TL-Bench 方法类别进行直接比较：

- **COMET Ref**（wmt22-comet-da）：基于参考译文，使用 gemini-2.5-flash 翻译作为参考译文
- **COMET QE**（wmt22-cometkiwi-da）：无参考质量估计
- **X-COMET-XL** [19]：基于参考译文，并提供可解释错误片段

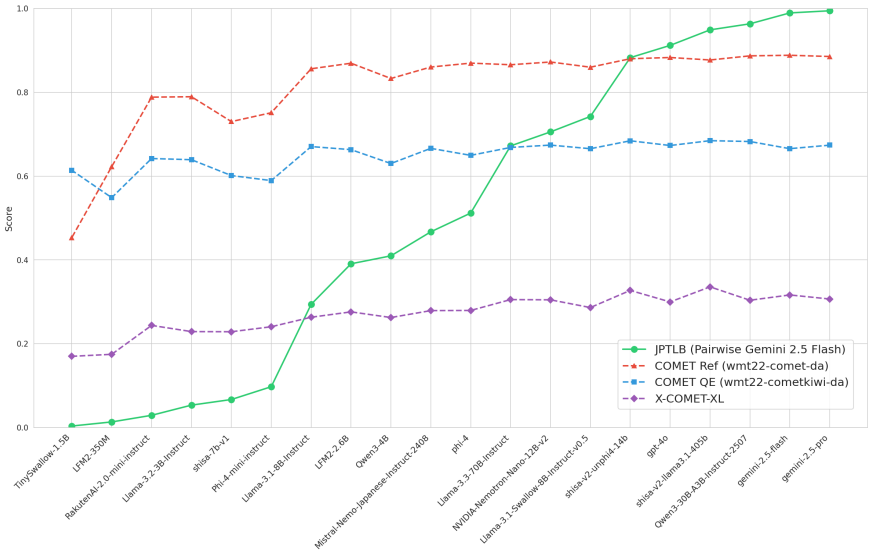

图 2： 按模型划分的分数变化。所有 20 个锚定模型的归一化分数（0–1 范围），按 JP-TL-Bench LT 分数排序。关键观察是顶端分数受到压缩：JP-TL-Bench 将排名前 6 的模型分布在有意义的范围内（LT 8.8–9.9），而 COMET 指标则将它们压缩在较窄的区间内（Ref 为 0.87–0.89，QE 为 0.66–0.68）。与 WMT22 模型相比，X-COMET-XL 的动态范围有所提升，但顶尖性能模型的分数仍集中在 0.30–0.34 的区间内。

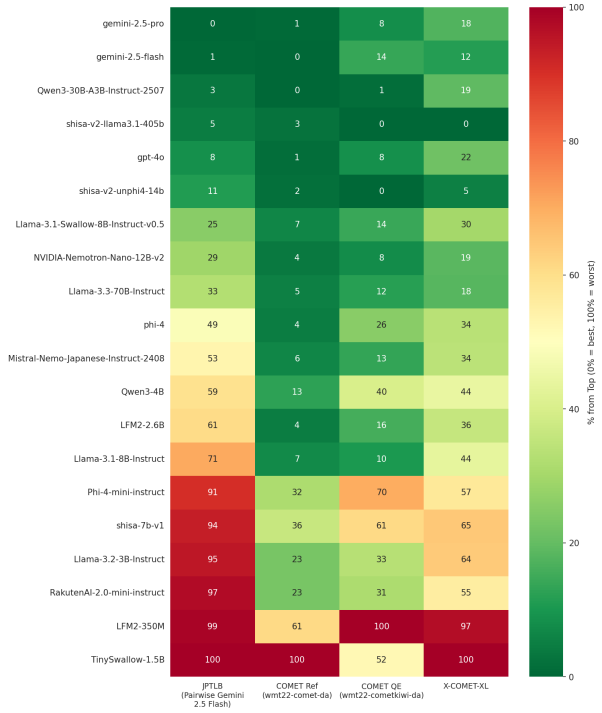

图 3： 相对位置热力图。各模型在每项指标中的位置以“距顶部的百分比”表示（0% ＝最佳，100% ＝最差）。COMET 指标将许多模型集中在 0–15% 区间，而 JP-TL-Bench 则在整个范围内呈现更渐进的区分效果——这正是开发阶段进行模型选择所需要的特性。

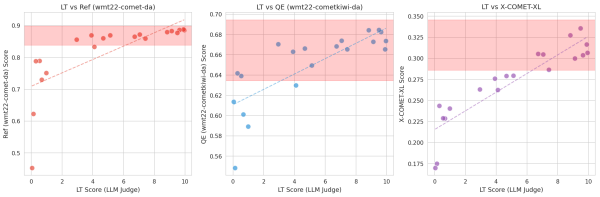

图 4： 饱和度散点图。红色阴影区域标出了这样一些位置：JP-TL-Bench 得分存在显著差异的模型，其 COMET 值却聚集在相近范围内。正向趋势线证实，COMET 指标能够正确反映质量变化方向，但纵向压缩限制了它们区分高水平系统的效用。

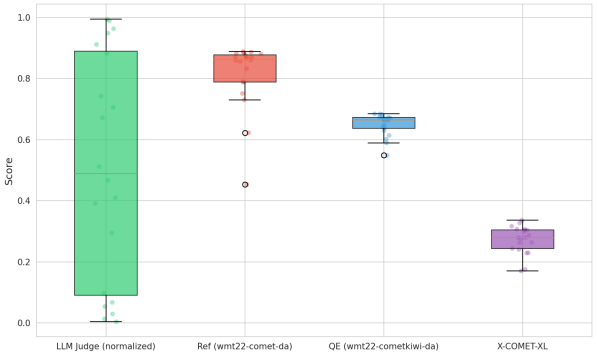

图 5： 分数分布比较。JP-TL-Bench（归一化后）的四分位距最大，能够最有效地区分模型。所有 COMET 变体的分数范围都明显更窄。

5.3 基于评分标准的评判

此外，我们还记录了 llm-jp-eval MT/LiquidAI 评估工作流（附录 C）中采用的基于评分标准的参考感知型大语言模型评判。评分标准评判器可提供 1–5 分制的可解释评分，并且相比基于 COMET 的评估器具有更好的区分度 [16]。

5.3.1 案例研究：LFM2-350M-ENJP-MT

Liquid AI 报告称：“LFM2-350M-ENJP-MT 所提供的翻译质量与规模大于它 10 倍以上的模型相当” [39], ，并展示了其 llm-jp-eval MT 分数图表：该分数不仅与 Gemma 3 4B 相当，也与 GPT-4o 相当（在规模上仅稍微大于 $10\times$ ）。通过适当的参数和提示模板调优，我们成功复现并确认了这些令人惊讶的 COMET 分数，但我们的定性分析表明，这种相当性是指标压缩造成的假象。

表 4： COMET 分数（llm-jp-eval MT）

模型	COMET 英语→日语	COMET日语→英语	COMET平均值
shisa-v2-llama3.1-405b	0.9165	0.8936	0.9050
GPT-4o	0.9212	0.8973	0.9093
LFM2-350M-ENJP-MT	0.9046	0.8731	0.8889
gemma-3-4b-it	0.8926	0.8694	0.8810

COMET 表格展示了顶端的压缩现象：LFM2-350M-ENJP-MT、Gemma 3 4B、GPT-4o 以及我们自己的 Shisa V2 405B 模型，得分几乎完全相同。通常，当人们看到这种分数平台时，普遍的假设是基准测试任务本身已经饱和（即翻译难度过低，无法区分不同模型）。然而，检查原始翻译后可以发现明显的质量差异。这表明存在**指标饱和**而非**任务饱和**：底层的性能差距确实存在，但 COMET 缺乏捕捉这种差距的分辨率。

应用基于评分标准的评判后，分数呈现出有意义的差异，能够更准确地反映可观察到的输出质量：

表5： 按模型划分的分数分布（基于评分标准的评判器）。百分比列表示各个1至5分评级的样本占比；有用率汇总了得分为3分及以上的样本；完美率表示得分为5分的样本占比。

模型	样本	Mean	中位数	1%	2%	3%	4%	5%	有用率	完美率
shisa-v2-llama3.1-405b	200	4.57	5.0	0.0	1.5	6.5	26.0	66.0	98.5	66.0
GPT-4o	200	4.55	5.0	0.0	0.5	7.5	28.0	64.0	99.5	64.0
LFM2-350M-ENJP-MT	200	3.96	4.0	0.0	12.5	18.5	29.5	39.5	87.5	39.5
gemma-3-4b-it	200	3.69	4.0	0.0	13.5	25.0	40.5	21.0	86.5	21.0

这一分布揭示了小型语言模型（SLM）的一项关键行为特征。尽管 350M 模型的 **“有用”评级高达 87.5%**（输出在概念上准确），但它在**“完美”指标（39.5%）**上的表现不佳，明显落后于前沿模型。关键在于，这种“ adequacy”并不能转化为偏好。在接受 JP-TL-Bench 的严格比较后，SLM 的排名大致符合对其能力的预期（附录 E）。这表明，尽管高度优化的 SLM 能够满足有效释义指标（COMET）和广泛的可接受性检查（有用率），但它们往往缺乏在与更大型模型的正面对比中胜出的文体细微差异和稳健性。它们实际上是**“正确但较差”**——只有成对判别才能可靠地捕捉到这种差异。

基于评分标准的评判很有用，近期也有许多工作致力于提升其质量[15, 16]。然而，绝对评分标准分数在高分段仍可能出现压缩，而且与相对比较相比，绝对评分本质上更容易受到评判者与提示词交互的影响。在我们的测试中，成对比较得出的排名比绝对评分标准评分更可靠、更一致，这也是 JP-TL-Bench 采用这一设计的原因（另见附录 C 和 D）。

6. 局限性与未来工作

1. **评判者依赖性**：分数取决于裁判模型和提示词；不同评判者可能得出不同的排序 [27, 13]。

2. **评判者自偏好**：默认评判者（gemini-2.5-flash）本身也是一个锚定模型，而已有研究表明，大语言模型评估器会偏好自身的输出以及相关模型的输出[40]。在我们对部分项目进行的双语抽查中，人工审阅者始终更偏好 Gemini 系列输出，这表明其排名靠前反映了真实的质量优势。尽管如此，在评估 Gemini 系列候选项时，用户可能更倾向于使用外部评判者，以尽量减少潜在的自偏好效应。

3. **覆盖范围**：70 个条目无法代表所有翻译领域（长上下文、对话翻译、专业法律/医学文本），但扩充默认测试集需要谨慎考虑，因为增加样本会导致成对比较数量成倍增长。针对每个新领域，适当的方法是使用替代测试集。

4. **单一语言对（当前）**：同样，该协议可推广到其他语言；但在扩展到新的语言对时，若要保持评估保真度，就需要整理新的锚点集，并确保胜率间距适当。

5. **可比范围**：只有在基准测试契约（基础集+评判者 +代码）下，分数才具有可比性。跨版本比较需要明确校准。

7. 结论

JP-TL-Bench 提供了一套基于锚点、由大语言模型评判的日语↔英语翻译评估协议，旨在增强开发阶段的区分能力。通过固定一个多样化的锚定集合，并使用 Bradley–Terry模型汇总比较结果， [1], 它能在明确的可复现性契约下生成稳定且易于解读的评分。与会将强大模型压缩到难以区分的区间内的 COMET 系列指标不同，JP-TL-Bench 能在整个质量范围内保持有意义的区分度。虽然我们专注于日语↔英语翻译，但基于锚点的成对比较方法也适用于其他语言对和评估领域，尤其是现有指标趋于饱和的场景——对于不断迭代提升强大大语言模型的质量而言，这正是最需要的特性。

附录 A：提示示例

A.1 简单提示（英→日）

目录：一段关于一名学生上学第一天的简短对话样例 ol.

示例：“昨天是我在河滨高中的第一天，哇，和以前的学校真是大不一样！校园非常大，有三栋独立的建筑物，还有一个令人惊叹的庭院，里面种满了樱花树。.....”

A.2 困难提示（英→日）

内容 **片段**：热门动作电子游戏《style id='3'>《合金装备崛起》中的一段对话 .

示例：“自由意志是个神话。宗教是个笑话。我们全都是棋子，受某种更强大的力量操控：模因。灵魂的 DNA。它们塑造我们的意志。它们就是文化——是我们传承的一切。让一个人长期暴露于愤怒之中，他就会学会仇恨。他会成为携带者。嫉妒、贪婪、绝望.....全都是模因。全都会代代相传。”

A.3 困难提示（日语→英语）

内容：《黑死馆杀人事件》的开篇（《黑死馆杀人事件》 / 《黑死馆杀人事件》）。这部作品被广泛认为是日语写就的最难读的书籍之一。

示例：「由于法水没有公开宣布圣阿列克谢寺院杀人案的侦破结果，到了谣传此案即将成为悬案的第十天，调查相关部门的首脑们从那天起不得不放弃追查拉扎列夫杀害者。那是因为，在降矢木宅邸——这座自四百年前便盘根错节、被称为臼杵耶稣会神学林以来神圣家族的宅邸中，突然开始出现一个如漆黑之风般四处徘徊的毒杀者.....」

附录 B：JP-TL-Bench 评审提示词（compare_prompt.txt）

这是 JP-TL-Bench 的 `translation_comparer_any_model.py`所使用的比较提示词：

```
You are an expert in evaluating translations between Japanese and English. Your task is to compare two translations
of the same source text and determine which one better captures the meaning, nuance, and natural flow of the original.

Here are the evaluation criteria:

1. Accuracy: Faithful representation of the source text's meaning2. Natural Expression: Fluent,
 idiomatic language in the target language3. Tone & Register: Appropriate formality level and style for the context
4. Cultural Adaptations: Appropriate handling of cultural references and idioms
5. Technical Precision: Accurate translation of specialized terms and concepts
6. Structural Flow: Natural sentence structure and paragraph organization7. Consistency: Uniform terminology and style throughout
8. Target Audience Consideration: Appropriateness for intended readersNote that if a translation is empty,
 it means a valid answer was not submitted and it loses the comparison by default.


Instructions:
1. Carefully read the source text and both translations
2. Review each translation against the evaluation criteria3. For each criterion,
 compare Translation A and Translation B
4. Determine which translation performed better for each criterion
5. Provide specific examples and brief explanations for your decisions
```

```
Complete your evaluation with:
<translation_analysis>
[Your detailed analysis of both translations, addressing each criterion with specific examples]
</translation_analysis>

<evaluation_summary>
[Brief summary of key differences and overall assessment]
</evaluation_summary>

<answer>(A or B ONLY, leave all commentary in translation_analysis. This will be machine graded, so only answer
with A or B here.)</answer>

{{formatted_data}}
```

附录 C：LiquidAI MT 评审器（llm-jp-eval 黑客松评分标准提示词）

在 LiquidAI 东京黑客松期间构建的工作流中，我们扩展了 llm-jp-eval MT，使其也能运行**绝对**、**参考译文感知型**、**基于评分标准的**大语言模型评审，以便与 COMET/BLEU 分数进行对比。此方法有助于研究高层次指标饱和现象，并且不同于 JP-TL-Bench（后者**不依赖参考译文**且采用**成对比较**）。

此分析的源代码见： https://github.com/lhl/liquid-ai-hackathon-tokyo/

C.1 评分标准（1–5 + 完全标记）

评判者按严格的 1–5 分制打分：

- 1：完全错误／未翻译／无法理解
- 2：重大错误，严重影响理解
- 3：尚可（传达了主要意思），但存在明显问题
- 4：良好（准确、自然），但有轻微瑕疵
- 5：优秀（达到专业水准）

此外，对于达到母语水准且毫无错误的译文，还会标记一个二元“完美”标志（`correct=1`）。

C.2 评审提示词模板（judge.j2）

这是我们评判者使用的模板（仅为排版而删节；内容保持不变）：

```
You are a bilingual expert evaluating machine translation quality between English and Japanese.

**Source text**: {{ source_text }}

**Translation**: {{ translated_text }}

**Reference translation** (gold standard): {{ reference_text }}
```

```
Evaluate the translation strictly on:- **Accuracy**: Does it preserve all facts, entities,
 and meaning from the source?- **Fluency**: Is it natural, grammatical,
 and idiomatic in the target language?
- **Completeness**: Is everything translated without omissions or hallucinations?

Use this 1-5 scoring rubric:
- **1** -- Completely wrong, untranslated, or incomprehensible- **2** -- Major errors in meaning, facts,
 or grammar that severely impact comprehension- **3** -- Adequate translation with the main idea conveyed,
 but noticeable issues in accuracy or fluency- **4** -- Good translation that is accurate and natural, with only minor imperfections
- **5** -- Excellent translation that is precise, fluent, and indistinguishable from a professional human translation


Additionally, mark **<correct>1</correct>** if the translation is perfect and would be acceptable as-is in a
professional setting (native-quality with no errors). Otherwise use **<correct>0</correct>**.

Respond with a brief justification followed by your scores:
<justification>...</justification><score>N</score><correct>0|1</correct>
```

附录 D：成对比较与数值评分

附录 C 介绍了一种基于评分标准的方法；本文将在此进一步说明 JP-TL-Bench 最终选择采用成对比较的原因。

原文（日语）：

> 兵库县有许多繁华时尚的街区，也拥有丰富的自然资源。县内有 41 个市町，县厅所在地神户市则分为中央区、东滩区等 9 个行政区。各个市町和区都有其独特的魅力。

Mistral 7B：

> 兵库县以其充满活力、时尚且自然资源丰富的城市而闻名。该县共有41个市町，首府神户市划分为九个行政区，例如中央区和东滩区。每个市、町或行政区都有其所在地区独特的魅力，不是吗?

Shisa Chotto：

> 兵库县拥有许多充满活力且时尚的城市，以及丰富的自然美景。该县由41个市町组成，其首府神户市划分为九个行政区，例如中央区和东滩区。每个市、町和行政区都有自己独特的魅力，不是吗?

上面比较了同一段简单日语文本的两种大语言模型生成的翻译。两者都准确，内容也基本相同。不过，Mistral 7B 存在一些轻微的语法错误（列举城市后用了逗号而非“和”），措辞略显奇怪，例如“受到大自然的丰厚眷顾”，还有一些过于直译的表达，例如“这片土地独有的魅力”。用数值分数来体现这些问题并不容易：在一段很长的文本中，单个逗号拼接错误该如何评分？又该如何判断“这片土地独有的”这样的短语是否别扭到需要扣分，如果需要，又该扣多少？100 分中扣 3 分？5 分?

这只是一个样本，但它体现了更广泛的原则：鉴于同一段文本可能有无穷多种翻译，成对比较能得出更加一致的评判结果。评估者很容易认同 Chotto 的翻译更胜一筹，即便他们给出的数值分数各不相同。这种一致性优势解释了为什么成对比较比绝对评分得出的排名更可靠（第 5.3 节）。通过进行数百次这样的比较，我们便能清晰了解某个模型在复杂的现实世界翻译任务中的相对实力。

附录 E：可复现性检查清单（JP-TL-Bench）

报告 JP-TL-Bench 得分时：

□ 基础集快照版本（例如， `baseset/v1.0`）□ 裁判模型标识符及提供方/版本（如适用）□ 裁判提示词版本（哈希值/路径）及解码设置（温度等）□ 候选模型标识符及解码设置□ 对缺失评审结果所做的任何过滤/回填□ 原始比较日志链接（或哈希值，如有）

附录 F：完整模型得分

供参考，我们列出开发过程中评估的部分模型在 JP-TL-Bench v1.0 上的得分：

EN→JA：英语到日语翻译

模型	简单 LT	困难 LT	总体 LT	胜率
google/gemini-3-flash-preview	9.98	10.00	9.99	97.3%
google/gemini-3-pro-preview	9.99	9.99	9.99	97.5%
google/gemini-2.5-pro	9.95	9.99	9.97	96.6%
google/gemini-2.5-flash	9.97	9.98	9.96	95.4%
deepseek-ai/DeepSeek-V3.1-Terminus	9.96	9.94	9.94	91.6%
openai/gpt-oss-120b	9.85	9.91	9.86	87.2%
moonshotai/Kimi-K2-Instruct-0905	9.86	9.90	9.86	87.5%
Qwen/Qwen3-235B-A22B-Instruct-2507	9.91	9.84	9.85	87.8%
google/gemma-3-27b-it	9.79	9.66	9.69	82.2%
shisa-ai/shisa-v2-llama3.1-405b	9.74	9.69	9.67	80.7%
shisa-ai/chotto	9.77	9.60	9.65	81.1%
shisa-ai/shisa-v2.1-unphi4-14b	9.59	9.74	9.63	81.6%
shisa-ai/shisa-v2.1-llama3.3-70b	9.50	9.79	9.63	80.4%
GPT-4o-2024-08-06	9.61	9.68	9.61	80.1%
Qwen/Qwen3-30B-A3B-Instruct-2507	9.48	9.72	9.56	79.9%
mistralai/Ministral-3-14B-Instruct-2512	9.43	9.71	9.55	78.9%
inclusionAI/Ling-1T-FP8	9.68	9.37	9.48	77.9%

模型	简单 LT	困难 LT	总体 LT	胜率
shisa-ai/shisa-v2-llama3.3-70b	9.08	9.64	9.36	76.0%
shisa-ai/shisa-v2-unphi4-14b	9.65	9.39	9.27	76.5%
shisa-ai/shisa-v2.1-qwen3-8b	9.23	9.34	9.21	75.1%
abeja/ABEJA-Qwen2.5-32b-Japanese-v1.0	9.24	9.08	9.07	72.2%
elyza/ELYZA-Shortcut-1.0-Qwen-32B	9.01	9.26	9.07	71.9%
tokyotech-llm/Llama-3.3-Swallow-70B-Instruct-v0.4	9.26	9.03	9.06	71.7%
stockmark/Stockmark-2-100B-Instruct	9.12	8.88	8.90	69.9%
elyza/ELYZA-Thinking-1.0-Qwen-32B	8.86	9.06	8.88	70.1%
tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.5	8.91	9.00	8.87	70.5%
unsloth/phi-4	8.87	8.75	8.70	68.4%
shisa-ai/shisa-v2.1-llama3.2-3b	8.48	9.01	8.68	68.4%
flux-inc/Flux-Japanese-Qwen2.5-32B-Instruct-V1.0	8.35	8.76	8.46	66.0%
nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16	7.16	7.32	7.10	58.5%
meta-llama/Llama-3.1-405B-Instruct	5.15	8.45	7.05	58.3%
shisa-ai/shisa-v2-llama3.1-8b	7.41	6.77	6.93	57.8%
nvidia/NVIDIA-Nemotron-Nano-12B-v2	7.27	6.85	6.87	58.2%
inclusionAI/Ling-flash-2.0	6.11	7.40	6.72	55.8%
Qwen/Qwen3-8B	6.85	6.86	6.72	57.9%
cyberagent/Mistral-Nemo-Japanese-Instruct-2408	6.62	6.62	6.45	56.2%
elyza/ELYZA-Shortcut-1.0-Qwen-7B	6.62	6.38	6.32	54.3%
shisa-ai/shisa-v2.1-lfm2-1.2b	6.66	5.30	5.75	51.6%
meta-llama/Llama-3.3-70B-Instruct	5.20	6.17	5.59	52.4%
LiquidAI/LFM2-8B-A1B	4.79	6.40	5.57	50.7%
LiquidAI/LFM2-2.6B	6.24	4.44	5.07	49.5%
openai/gpt-oss-20b	8.35	2.41	4.92	48.7%
sbintuitions/sarashina2.2-3b-instruct-v0.1	4.46	5.41	4.86	48.0%
microsoft/phi-4	3.80	4.64	4.16	46.1%
mistralai/Ministral-3-3B-Instruct-2512	3.46	4.52	3.95	44.2%
meta-llama/Llama-4-Scout-17B-16E	1.92	5.41	3.67	43.2%
baidu/ERNIE-4.5-21B-A3B-PT	3.11	3.71	3.42	41.9%
Qwen/Qwen3-4B	1.79	3.62	2.70	39.3%
meta-llama/Llama-3.2-1B-Instruct	1.24	2.44	1.88	30.7%
LiquidAI/LFM2-1.2B	2.07	1.27	1.63	33.2%
baidu/ERNIE-4.5-VL-28B-A3B-PT	1.36	1.46	1.47	30.6%
meta-llama/Llama-3.1-8B-Instruct	1.01	1.66	1.37	31.2%
microsoft/Phi-4-mini-instruct	0.77	1.07	0.97	28.0%
Nanbeige/Nanbeige4-3B-Thinking-2511	0.58	1.04	0.87	25.0%
augmxnt/shisa-7b-v1	0.91	0.46	0.70	24.7%
augmxnt/shisa-gamma-7b-v1	0.76	0.32	0.54	20.7%
openbmb/MiniCPM4.1-8B	0.35	0.62	0.54	21.0%
microsoft/Phi-4-multimodal-instruct	1.09	0.17	0.52	20.8%
meta-llama/Llama-3.2-3B-Instruct	0.19	0.35	0.32	17.2%
allenai/Olmo-3-7B-Instruct	0.15	0.30	0.28	16.5%
Rakuten/RakutenAI-2.0-mini-instruct	0.65	0.05	0.25	15.3%
LiquidAI/LFM2-350M	0.08	0.08	0.12	11.8%
mistralai/Mistral-7B-Instruct-v0.1	0.08	0.06	0.09	8.9%
SakanaAI/TinySwallow-1.5B	0.01	0.01	0.01	1.4%

JA→EN：日译英翻译

模型	简单 LT	困难 LT	总体 LT	胜率
google/gemini-3-flash-preview	9.94	9.99	9.98	97.4%
google/gemini-3-pro-preview	9.88	9.98	9.94	94.1%
google/gemini-2.5-pro	9.79	9.99	9.94	95.7%

模型	简单 LT	困难 LT	总体 LT	胜率
Qwen/Qwen3-235B-A22B-Instruct-2507	9.83	9.97	9.93	93.2%
openai/gpt-oss-120b	9.89	9.95	9.92	92.6%
deepseek-ai/DeepSeek-V3.1-Terminus	9.85	9.93	9.89	90.9%
inclusionAI/Ling-1T-FP8	9.93	9.89	9.89	91.0%
mistralai/Ministral-3-14B-Instruct-2512	9.90	9.89	9.88	90.7%
moonshotai/Kimi-K2-Instruct-0905	9.79	9.92	9.86	89.3%
google/gemini-2.5-flash	9.83	9.88	9.84	90.4%
shisa-ai/chotto	9.96	9.70	9.79	87.4%
Qwen/Qwen3-30B-A3B-Instruct-2507	9.70	9.84	9.77	86.5%
shisa-ai/shisa-v2-llama3.3-70b	9.74	9.83	9.77	85.9%
shisa-ai/shisa-v2-llama3.1-405b	9.65	9.83	9.74	83.5%
google/gemma-3-27b-it	9.54	9.77	9.66	82.7%
shisa-ai/shisa-v2.1-llama3.3-70b	9.43	9.80	9.66	82.5%
GPT-4o-2024-08-06	9.59	9.74	9.65	82.4%
shisa-ai/shisa-v2.1-unphi4-14b	9.55	9.28	9.36	78.5%
shisa-ai/shisa-v2-unphi4-14b	9.44	9.20	9.17	76.5%
shisa-ai/shisa-v2.1-qwen3-8b	8.77	9.26	9.02	73.7%
inclusionAI/Ling-flash-2.0	8.85	9.19	9.00	72.3%
abeja/ABEJA-Qwen2.5-32b-Japanese-v1.0	8.36	9.08	8.74	69.3%
unsloth/phi-4	8.61	8.90	8.72	69.2%
nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16	9.47	8.00	8.69	70.1%
tokyotech-llm/Llama-3.3-Swallow-70B-Instruct-v0.4	8.15	8.89	8.57	68.1%
elyza/ELYZA-Shortcut-1.0-Qwen-32B	8.04	8.94	8.53	67.1%
meta-llama/Llama-3.1-405B-Instruct	9.05	8.12	8.49	67.0%
elyza/ELYZA-Thinking-1.0-Qwen-32B	7.73	8.80	8.34	65.6%
shisa-ai/shisa-v2-llama3.1-8b	8.10	8.56	8.30	65.2%
meta-llama/Llama-3.3-70B-Instruct	8.46	8.09	8.19	65.5%
stockmark/Stockmark-2-100B-Instruct	8.57	7.97	8.18	64.7%
openai/gpt-oss-20b	8.47	7.96	8.12	65.3%
Qwen/Qwen3-8B	6.95	8.63	7.94	64.6%
flux-inc/Flux-Japanese-Qwen2.5-32B-Instruct-V1.0	7.32	8.09	7.75	62.2%
nvidia/NVIDIA-Nemotron-Nano-12B-v2	7.93	7.25	7.49	61.7%
mistralai/Ministral-3-3B-Instruct-2512	7.97	6.88	7.29	58.2%
shisa-ai/shisa-v2.1-llama3.2-3b	8.50	5.86	7.09	57.3%
baidu/ERNIE-4.5-21B-A3B-PT	6.28	7.69	7.07	57.2%
tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.5	6.25	7.10	6.69	56.5%
microsoft/phi-4	5.54	6.38	6.00	53.6%
baidu/ERNIE-4.5-VL-28B-A3B-PT	6.12	5.87	5.94	51.7%
Qwen/Qwen3-4B	5.38	5.36	5.34	50.4%
elyza/ELYZA-Shortcut-1.0-Qwen-7B	4.85	5.21	5.01	46.9%
sbintuitions/sarashina2.2-3b-instruct-v0.1	4.66	4.93	4.79	46.0%
meta-llama/Llama-3.1-8B-Instruct	6.07	3.74	4.70	47.1%
meta-llama/Llama-4-Scout-17B-16E	4.82	3.17	3.84	41.9%
LiquidAI/LFM2-8B-A1B	4.17	2.80	3.37	39.3%
LiquidAI/LFM2-2.6B	4.06	2.38	3.06	38.8%
cyberagent/Mistral-Nemo-Japanese-Instruct-2408	3.37	2.55	2.91	38.8%
Nanbeige/Nanbeige4-3B-Thinking-2511	4.29	1.93	2.87	36.5%
openbmb/MiniCPM4.1-8B	3.86	1.79	2.57	35.5%
microsoft/Phi-4-multimodal-instruct	2.72	1.18	1.75	30.8%
allenai/Olmo-3-7B-Instruct	1.92	1.25	1.57	29.6%
shisa-ai/shisa-v2.1-lfm2-1.2b	2.89	0.68	1.37	28.0%
microsoft/Phi-4-mini-instruct	2.22	0.82	1.31	29.3%
meta-llama/Llama-3.2-3B-Instruct	0.94	0.61	0.80	24.9%
augmxnt/shisa-7b-v1	0.82	0.45	0.63	22.6%
augmxnt/shisa-gamma-7b-v1	0.99	0.35	0.61	21.2%
LiquidAI/LFM2-1.2B	0.83	0.27	0.48	19.6%
mistralai/Mistral-7B-Instruct-v0.1	0.59	0.15	0.30	14.8%

模型	简单 LT	困难 LT	总体 LT	胜率
Rakuten/RakutenAI-2.0-mini-instruct	0.42	0.12	0.23	13.1%
meta-llama/Llama-3.2-1B-Instruct	0.13	0.11	0.14	9.9%
LiquidAI/LFM2-350M	0.08	0.08	0.12	11.8%
SakanaAI/TinySwallow-1.5B	0.03	0.04	0.05	4.0%

参考资料

[1] Ralph Allan Bradley 和 Milton E Terry。不完全区组设计的秩分析：I. 成对比较法。*Biometrika*，39(3/4):324–345，1952。

[2] Shisa.AI。Shisa v2：双语日英大语言模型。 https://shisa.ai/posts/shisa-v2/，2025。

[3] Shisa.AI。Shisa V2 405B。 https://shisa.ai/posts/shisa-v2-405b/，2025。

[4] Shisa.AI。Shisa v2.1 发布。 https://shisa.ai/posts/shisa-v2.1/，2025。

[5] LLM-jp 项目。llm-jp-eval：日语大语言模型的自动评估工具。 https://github.com/llm-jp/llm-jp-eval，2024。

[6] Kishore Papineni、Salim Roukos、Todd Ward 和 Wei-Jing Zhu。BLEU：一种机器翻译自动评估方法。载于计算语言学协会第*40*届年会论文集，第311–318页，美国宾夕法尼亚州费城，2002年7月。计算语言学协会。

[7] Maja Popović。chrF：用于机器翻译自动评估的字符 n 元语法 F 分数。载于 Ondřej Bojar、Rajan Chatterjee、Christian Federmann、Barry Haddow、Chris Hokamp、Matthias Huck、Varvara Logacheva 和 Pavel Pecina 编，第十届统计机器翻译研讨会论文集，第392–395页，葡萄牙里斯本，2015年9月。计算语言学协会。

[8] Ricardo Rei、Craig Stewart、Ana C Farinha 和 Alon Lavie。COMET：一种用于机器翻译评估的神经网络框架。载于*2020*年自然语言处理实证方法会议（*EMNLP*）论文集，第2685–2702页，线上，2020年11月。计算语言学协会。

[9] Sweta Agrawal、António Farinhas、Ricardo Rei 和 Andre Martins。自动指标能否评估高质量译文？载于 Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen 编*2024*年自然语言处理实证方法会议论文集，第14491–14502页，美国佛罗里达州迈阿密，2024年11月。计算语言学协会。

[10] Vilém Zouhar、Pinzhen Chen、Tsz Kin Lam、Nikita Moghe 和 Barry Haddow。使用 COMET 的陷阱与展望，2024。

[11] Ehud Reiter。《BLEU有效性的结构化评审》。《计算语言学》，44(3)：393–401，2018年。

[12] 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 和 Ion Stoica。使用 MT-Bench 和 Chatbot Arena 评判大语言模型裁判，2023年。NeurIPS 2023 数据集与基准赛道（亦发表于 OpenReview）。[13] Jiawei Gu、Xuhui Jiang、Zhichao Shi、Hexiang Tan、Xuehao Zhai、Chengjin Xu、Wei Li、Yinghan Shen、Shengjie Ma、Honghao Liu、Saizhuo Wang、Kun Zhang、Yuanzhuo Wang、Wen Gao、Lionel Ni 和 Jian Guo。大语言模型裁判综述，2024年。[14] Wei-Lin Chiang、Lianmin Zheng、Ying Sheng、Anastasios Nikolas Angelopoulos、Tianle Li、Dacheng Li、Hao Zhang、Banghua Zhu、Michael I. Jordan、Joseph E. Gonzalez 和 Ion Stoica。Chatbot Arena：一个通过人类偏好评估大语言模型的开放平台，2024年。[15] Seungone Kim、Jamin Shin、Yejin Cho、Joel Jang、Shayne Longpre、Hwaran Lee、Sangdoo Yun、Seongjin Shin、Sungdong Kim、James Thorne 和 Minjoon Seo。Prometheus：诱导语言模型具备细粒度评估能力，2024年。[16] Ganqu Cui、Lifan Yuan、Ning Ding、Guanming Yao、Bingxiang He、Wei Zhu、Yuan Ni、Guotong Xie、Ruobing Xie、Yankai Lin、Zhiyuan Liu 和 Maosong Sun。UltraFeedback：通过规模化人工智能反馈提升语言模型，2024年。[17] Markus Freitag、Ricardo Rei、Nitika Mathur、Chi-kiu Lo、Craig Stewart、Eleftherios Avramidis、Tom Kocmi、George Foster、Alon Lavie 和 André F. T. Martins。WMT22 指标共享任务结果：停止使用 BLEU——神经指标更优且更稳健。载于《第七届机器翻译会议论文集（*WMT*）》，第46–68页，阿布扎比，阿拉伯联合酋长国（混合形式），2022年12月。计算语言学协会。[18] Ricardo Rei、Marcos Treviso、Nuno M. Guerreiro、Chrysoula Zerva、Ana C. Farinha、Christine Maroti、José G. C. de Souza、Taisiya Glushkova、Duarte M. Alves、Alon Lavie、Luisa Coheur 和 André F. T. Martins。COMETKiwi：IST-Unbabel 2022年质量估计共享任务提交论文，2022年。[19] Nuno M. Guerreiro、Ricardo Rei、Daan van Stigt、Luisa Coheur、Pierre Colombo 和 André F. T. Martins。xCOMET：通过细粒度错误检测实现透明的机器翻译评估，2023年。[20] Arle Lommel、Hans Uszkoreit 和 Aljoscha Burchardt。多维质量指标（MQM）：用于声明和描述翻译质量指标的框架。*Tradumàtica:tecnologies de la traducció*，（12）：455–463，2014年。[21] Yvette Graham、Timothy Baldwin、Alistair Moffat 和 Justin Zobel。机器翻译人工评估中的连续测量尺度。载于 Antonio Pareja-Lora、Maria Liakata 和 Stefanie Dipper（编），《第七届语言标注研讨会与话语互操作性论文集》，第33–41页，索非亚，保加利亚，2013年8月。计算语言学协会。[22] Markus Freitag、George Foster、David Grangier、Viresh Ratnakar、Qijun Tan 和 Wolfgang Macherey。专家、错误与语境：机器翻译人工评估的大规模研究，2021年。[23] Tom Kocmi 和 Christian Federmann。大语言模型是翻译质量评估的最先进评估器，2023年。

[24] Tom Kocmi 和 Christian Federmann。GEMBA-MQM：使用 GPT-4 检测翻译质量错误片段。载于 Philipp Koehn、Barry Haddow、Tom Kocmi 和 Christof Monz 编，《第八届机器翻译会议论文集》，第 768–775 页，新加坡，2023年12月。计算语言学协会。[25] Daniil Larionov 和 Steffen Eger。BatchGEMBA：通过批量提示和提示压缩实现令牌高效的机器翻译评估，2025。[26] Kayla Schroeder 和 Zach Wood-Doughty。你能相信大语言模型的判断吗？大语言模型裁判的可靠性，2024。[27] Peiyi Wang、Lei Li、Liang Chen、Zefan Cai、Dawei Zhu、Binghuai Lin、Yunbo Cao、Qi Liu、Tianyu Liu 和 Zhifang Sui。大语言模型并非公平的评估器，2023。[28] Miyoung Ko、Jinhyuk Lee、Hyunjae Kim、Gangwoo Kim 和 Jaewoo Kang。关注首句：问答中的位置偏差，2020。[29] Arpad E. Elo。《国际象棋选手评级：过去与现在》。Arco Publishing，1978年。[30]匿名作者。FiRE：面向机器翻译的细粒度排序评估。https://openreview.net/forum?id=4mpOLUAcHD，2026。正在国际学习表征会议 ICLR 2026 作为会议论文接受同行评审。PDF： https://openreview.net/pdf?id=4mpOLUAcHD。[31] Naman Goyal、Cynthia Gao、Vishrav Chaudhary、Peng-Jen Chen、Guillaume Wenzek、Da Ju、Sanjana Krishnan、Marc'Aurelio Ranzato、Francisco Guzmán 和 Angela Fan。面向低资源和多语机器翻译的 FLORES-101 评估基准。《计算语言学协会汇刊》，10：522–538，2022。[32] NLLB团队、Marta R. Costa-jussà、James Cross、Onur Çelebi、Maha Elbayad、Kenneth Heafield、Kevin Heffernan、Elahe Kalbassi、Janice Lam、Daniel Licht、Jean Maillard、Anna Sun、Skyler Wang、Guillaume Wenzek、Al Youngblood、Bapi Akula、Loic Barrault、Gabriel Mejia Gonzalez、Prangthip Hansanti、John Hoffman、Semarley Jarrett、Kaushik Ram Sadagopan、Dirk Rowe、Shannon Spruit、Chau Tran、Pierre Andrews、Necip Fazil Ayan、Shruti Bhosale、Sergey Edunov、Angela Fan、Cynthia Gao、Vedanuj Goswami、Francisco Guzmán、Philipp Koehn、Alexandre Mourachko、Christophe Ropers、Safiyyah Saleem、Holger Schwenk 和 Jeff Wang。不让任何语言掉队：扩展以人为中心的机器翻译，2022。[33] Makoto Morishita、Jun Suzuki 和 Masaaki Nagata。JParaCrawl：大规模基于网络的英日平行语料库。载于《第十二届语言资源与评测会议论文集》，第 3603–3609 页，法国马赛，2020年5月。欧洲语言资源协会。[34] Longyue Wang、Zhaopeng Tu、Shuming Shi、Tong Zhang、Yvette Graham 和 Qun Liu。使用重构模型翻译代词省略语言。载于 AAAI人工智能会议论文集，第 32 卷，第 4937–4945 页，2018。[35] Sho Shimazu、Sho Takase、Toshiaki Nakazawa 和 Naoaki Okazaki。日译英中的日语零代词评估数据集。载于《第十二届语言资源与评测会议论文集》，第 3630–3634 页，法国马赛，2020年5月。欧洲语言资源协会。

[36] Weston Feely、Eva Hasler 和 Adrià de Gispert。控制英译日神经机器翻译中的日语敬语。载于第六届亚洲翻译研讨会，第 45–53 页，中国香港，2019年11月。计算语言学协会。[37] Pat Verga、Sebastian Hofstatter、Sophia Althammer、Yixuan Su、Aleksandra Piktus、Arkady Arkhangorodsky、Minjie Xu、Naomi White 和 Patrick Lewis。以评审团取代评判者：使用多样化模型评审团评估大语言模型生成内容，2024。[38] Lucas Maystre。choix：基于Luce选择公理的模型推断算法。 https://github.com/lucasmaystre/choix，2015年。[39] Liquid AI。LFM2-350M-ENJP-MT 模型卡。https://huggingface.co/LiquidAI/LFM2-350M-ENJP-MT，2025年。[40] Arjun Panickssery、Samuel R. Bowman 和 Shi Feng。大语言模型评估器能够识别并偏爱自身生成的内容，2024。