拉加斯：检索增强生成的自动化评估

Shahul Es[†], **Jithin James**[†], **Luis Espinosa-Anke**[*◇], **Steven Schockaert**[*]
[†]Exploding Gradients
[*]CardiffNLP, Cardiff University, United Kingdom
[◇]AMPLYFI, United Kingdom
shahules786@gmail.com,jamesjithin97@gmail.com
{espinosa-ankel,schockaerts1}@cardiff.ac.uk

Abstract

我们介绍了一个名为**拉加斯**（**R** 检索 **A** 增强 **G** 生成**作为**评估）的框架，用于对检索增强生成 (RAG) 管道进行无参考评估。RAG 系统由一个检索模块和一个基于 LLM 的生成模块组成，并为 LLMs 提供来自参考文本数据库的知识，使其能够作为用户与文本数据库之间的自然语言层，降低幻觉的风险。然而，评估 RAG 架构具有挑战性，因为有多个方面需要考虑：检索系统识别相关且聚焦的上下文片段的能力，LLM 利用这些片段的忠实程度，或者生成本身的质量。通过 Ragas，我们提出了一套可以用来评估这些不同方面的指标*无需依赖地面真实的人工标注*。我们认为这样的框架对于加快 RAG 架构的评估周期至关重要，在 LLMs 迅速被采用的情况下这一点尤为重要。

1 介绍

语言模型（LMs）捕获了关于世界的大量知识，这使得它们能够在不访问任何外部资源的情况下回答问题。将 LMs 视为知识库的想法在 BERT(Devlin et al., 2019) 引入后不久出现，并随着更大规模的 LMs(Roberts et al., 2020) 的引入而更加稳固。尽管最近的大语言模型（LLMs）捕获的知识足以在各种问答基准测试 (Bubeck et al., 2023) 中与人类表现相匹敌，但将 LLMs 用作知识库的想法仍然存在两个基本限制。首先，LLMs 无法回答在其训练后发生事件的问题。其次，即使是最大的模型也难以记住仅在训练语料库中很少提及的知识 (Kandpal et al., 2022; Mallen et al., 2023)。解决这些问题的标准方法是依赖于*检索增强生成*（*RAG*）(Lee et al., 2019; Lewis et al., 2020; Guu et al., 2020) 。然后回答问题基本上涉及从语料库中检索相关段落，并将这些段落与原始问题一起提供给 LM。虽然最初的方案依赖于专门的用于检索增强语言模型的 LMs (Khandelwal et al., 2020; Borgeaud et al., 2022)，近期的研究表明，只需将检索到的文档添加到标准 LM 输入中也可以很好地工作 (Khattab et al., 2022; Ram et al., 2023; Shi et al., 2023)，从而使得可以将检索增强策略与仅通过 API 可用的 LLMs 结合使用。

尽管检索增强策略的实用性显而易见，但其实施需要大量的调优，因为整体性能将受到检索模型、考虑的语料库、语言模型或提示格式等因素的影响。因此，对检索增强系统的自动化评估至关重要。实际上，RAG 系统通常根据语言建模任务本身进行评估，即通过在某个参考语料库上测量困惑度来衡量。然而，这样的评估并不总是能够预测下游性能 (Wang et al., 2023c)。此外，这种评估策略依赖于语言模型的概率，而这些概率对于某些封闭模型（如 ChatGPT 和 GPT-4）是无法访问的。问答是另一个常见的评估任务，但通常只考虑具有简短提取答案的数据集，这可能并不能代表系统将如何被使用。

为了解决这些问题，在本文中我们提出了 Ragas[1]，一个用于检索增强生成系统的自动化评估框架。我们关注的是参考答案可能不可用的情况，并且我们想要估计不同正确性的代理指标，以及检索到的段落的有用性。Ragas 框架提供了与 llama-index 和语言链这两种最常

[1]拉加斯可在 https://github.com/explodinggradients/ragas 获取。

用的构建 RAG 解决方案框架的集成，从而使开发人员能够轻松地将 Ragas 集成到他们的标准工作流程中。

2 相关工作

使用大语言模型估计保真度 检测大型语言模型生成响应中的幻觉问题已被广泛研究 (Ji et al., 2023)。几位作者提出了使用少量示例提示策略预测事实性的想法 (Zhang et al., 2023)。然而，最近的分析表明，现有的模型在使用标准提示策略时难以检测到幻觉 (Li et al., 2023; Azaria and Mitchell, 2023)。其他方法依赖于将生成的响应与外部知识库中的事实联系起来 (Min et al., 2023)，但这并不总是可能的。

另一种策略是检查分配给单个标记的概率，在此情况下，我们预期模型对虚构答案的自信程度会低于对事实性答案的自信。例如，BARTScore(Yuan et al., 2021) 通过查看生成文本在给定输入条件下的概率来评估其事实性。Kadavath et al. (2022) 使用这一想法的一个变体。从大型语言模型在回答多项选择题时提供校准良好的概率这一点出发，他们基本上将验证模型生成答案的问题转化为一个多项选择问题，询问该答案是真是假。而不是查看输出概率，Azaria and Mitchell (2023) 提出训练一个监督分类器来预测给定陈述是否真实，这个分类器基于大型语言模型隐藏层之一的权重进行训练。尽管这种方法表现良好，但需要访问模型的隐藏状态使其不适合通过 API 访问大型语言模型的系统。

对于不提供访问令牌概率的模型，如 ChatGPT 和 GPT-4，需要不同的方法。SelfCheckGPT(Manakul et al., 2023) 通过采样多个答案来解决这个问题。他们的核心思想是事实性答案更稳定：当一个答案是事实性的，我们可以期望不同的样本在语义上会趋向相似，而对于幻觉产生的答案则不太可能如此。

文本生成系统的自动评估 大型语言模型也被用来自动评估生成文本片段的其他方面，而不仅仅是事实性。例如，GPTScore(Fu et al., 2023) 使用一个指定考虑方面的提示（如流利度），然后根据给定自回归 LM 生成标记的平均概率对段落进行评分。这种使用提示的想法之前也由 Yuan et al. (2021) 考虑过，尽管他们使用了一个较小的微调模型（即 BART）并且没有观察到使用提示带来的明显好处。另一种方法直接要求 ChatGPT 通过提供 0 到 100 之间的分数或在 5 星评级系统中提供评分来评估给定答案的特定方面 (Wang et al., 2023a)。令人惊讶的是，可以以这种方式获得强大的结果，尽管它受到提示设计敏感性的限制。而不是对个别答案进行评分，一些作者也专注于使用大型语言模型从多个候选答案中选择最佳答案 (Wang et al., 2023b)，通常是为了比较不同大型语言模型的表现。然而，这种方法需要注意，因为呈现答案的顺序可能会对结果产生影响 (Wang et al., 2023b)。

在如何使用真实答案或更广泛地说，生成内容方面，大多数方法依赖于有一个或多个参考答案的可用性。例如，BERTScore (Zhang et al., 2020) 和 MoverScore (Zhao et al., 2019) 使用预训练 BERT 模型产生的上下文化嵌入来比较生成的答案与参考答案之间的相似度。BARTScore (Yuan et al., 2021) 类似地使用参考答案来计算诸如精度（估计为给定参考答案时生成生成答案的概率）和召回率（估计为给定生成答案时生成参考答案的概率）等方面。

3 评估策略

我们考虑一个标准的 RAG 设置，在该设置中，给定一个问题 q，系统首先检索一些上下文 $c(q)$，然后使用检索到的上下文生成答案 $a_s(q)$。在构建 RAG 系统时，我们通常无法访问人工标注的数据集或参考答案。因此，我们专注于完全自包含且无需参考的指标。我们特别关注三个质量方面，我们认为这些方面至关重要。首先，**忠实性**指的是答案应该基于给定上下文的想法。这是为了避免产生幻想，并确保检索到的上下文可以作为生成答案的依据。

事实上，在法律等领域中，RAG 系统常用于应用场合，其中生成文本的事实一致性与所依据的来源高度相关，信息在这些领域不断演变。其次，**答案相关性**指的是生成的答案应解决实际提供的问题。最后，**上下文相关性**指的是检索到的上下文应该集中，包含尽可能少的相关性不强的信息。鉴于将长篇幅的上下文输入大语言模型的成本，这一点非常重要。此外，当上下文段落过长时，大语言模型往往不太能有效地利用这些上下文，特别是对于那些出现在上下文中段的信息 (Liu et al., 2023)。

我们现在解释如何可以通过提示大型语言模型（LLM）以完全自动的方式测量这三个质量方面。在我们的实现和实验中，所有提示都使用通过 OpenAI API[2] 可获得的 `gpt-3.5-turbo-16k` 模型进行评估。

忠实性　我们说答案 $a_s(q)$ 对上下文 $c(q)$ 是忠实的，如果答案中提出的主张可以从上下文中推断出来。为了估计忠实度，我们首先使用一个大语言模型来提取一组陈述，$S(a_s(q))$。这一步的目标是将较长的句子分解成更短且更具针对性的断言。我们使用以下提示进行此步骤[3]：

> 给定一个问题和答案，从给出的答案中的每个句子创建一个或多个陈述。
> 问题：[问题]
> 答案：[答案]

其中 [问题] 和 [答案] 指代给定的问题和答案。对于每个陈述 s_i 在 S 中，大语言模型确定是否可以使用验证函数 $v(s_i, c(q))$ 从 $c(q)$ 推断出 s_i。此验证步骤使用以下提示进行：

> 考虑给定的背景信息和以下陈述，然后确定这些陈述是否得到了背景信息的支持。在得出裁决（是/否）之前为每个陈述提供简要解释。按照给定格式在末尾依次给出每个陈述的最终裁决。不要偏离指定的格式。
> 陈述：[陈述 1]
> ...
> 陈述：[声明 n]

最后的忠实度得分，F，则计算为 $F = \frac{|V|}{|S|}$，其中 $|V|$ 是根据 LLM 支持的声明数量，而 $|S|$ 则是声明总数。

答案相关性　我们说答案 $a_s(q)$ 是相关的，如果它直接且恰当地解决了问题。特别是，我们对答案相关性的评估不考虑事实性，但会惩罚那些答案不完整或包含冗余信息的情况。为了估计答案的相关性，对于给定的答案 $a_s(q)$，我们提示大语言模型生成基于 $a_s(q)$ 的 n 个潜在问题 q_i，如下所示：

> 生成一个与给定答案相关的问题。
> 答案：[答案]

然后，我们使用来自 OpenAI API 的 `text-embedding-ada-002` 模型为所有问题获取嵌入。对于每个 q_i，我们计算其与原始问题 q 的相似度 $\text{sim}(q, q_i)$，即相应的嵌入之间的余弦值。然后，问题 q 的答案相关性得分 AR 计算如下：

$$\text{AR} = \frac{1}{n}\sum_{i=1}^{n}\text{sim}(q, q_i) \tag{1}$$

此指标评估生成的答案与初始问题或指令的吻合程度。

上下文相关性　上下文 $c(q)$ 被认为是相关的，因为它仅包含回答问题所需的信息。特别是，这一指标旨在惩罚冗余信息的加入。为了估计上下文的相关性，给定一个问题 q 及其上下文 $c(q)$，大语言模型从 $c(q)$ 中提取出一组关键句子 S_{ext}，用于回答 q，使用以下提示：

> 请从提供的上下文中提取可能有助于回答以下问题的相关句子。如果未找到相关句子，或者你认为给定的上下文无法回答该问题，则返回短语“信息不足”。在抽取候选句子时，不允许对给定上下文中的句子进行任何更改。

[2] https://platform.openai.com

[3] 为了澄清任务，我们作为提示的一部分包含了一个演示。这个演示在本文中的提示列表中没有明确显示。

然后计算上下文相关性分数为：

$$\text{CR} = \frac{\text{number of extracted sentences}}{\text{total number of sentences in } c(q)} \quad (2)$$

4 WikiEval 数据集

为了评估所提出的框架，我们理想上需要问题-上下文-答案三元组的示例，并且这些示例需要附带人类判断。然后我们可以验证我们的指标在多大程度上与人类对忠实性、答案相关性和上下文相关性的评估一致。由于我们不知道有任何公开可用的数据集可以用于此目的，我们创建了一个新数据集，我们称之为维基评估[4]。为了构建该数据集，我们首先选择了50个维基百科页面，这些页面涵盖了自2022年[5]以来发生的事件。在选择这些页面时，我们优先考虑那些最近有编辑的页面。对于这50个页面中的每一个，我们都要求ChatGPT提出一个可以根据页面简介部分回答的问题，使用以下提示：

> 你的任务是从给定的上下文中制定一个问题，满足以下规则：
> *1.* 问题应能从给定的上下文中完全回答。
> *2.* 问题应基于包含非平凡信息的部分来构建。
> *3.* 答案不应包含任何链接。
> *4.* 问题难度应适中。
> *5.* 问题必须合理，并且人类能够理解和回应。
> *6.* 在问题中不要使用“提供的上下文”等短语。
> 上下文：

我们也使用ChatGPT来回答生成的问题，在给定相应的简介部分作为上下文的情况下，使用以下提示：

> 回答问题时使用给定上下文中的信息。
> 问题：[问题]
> 上下文：[上下文]

所有问题都由两位标注者按照三个考虑的质量维度进行了标注。这两位标注者都能流利地使用英语，并且他们收到了关于这三个考虑的质量维度含义的明确指示。对于忠实性和上下文相关性，两名标注者在大约95%的情况下达成了一致意见。对于答案的相关性，他们在大约90%的情况下达成一致。分歧是在标注者之间讨论后解决的。

忠实性 为了获得关于忠实度的人类判断，我们首先使用ChatGPT在没有访问任何额外上下文的情况下回答问题。然后我们要求标注者根据问题和相应维基百科页面来判断两个答案中哪个最忠实（即标准答案或无上下文生成的答案）。

答案相关性 我们首先使用ChatGPT获取具有较低答案相关性的候选答案，使用的提示如下：

> 回答给定的问题但不完整。
> 问题：[问题]

然后我们请人工标注者比较这个答案，并指出两个答案中哪个具有最高的答案相关性。

上下文相关性 为了衡量这一方面，我们首先通过抓取相应维基百科页面的回链来为上下文添加额外的句子。这样，我们就能够向上下文中添加相关但对回答问题不太相关的资讯。对于没有回链的少数页面，我们则使用ChatGPT来完成给定的上下文。

5 实验

表1分析了第3节中提出的指标与从所提出的WikiEval数据集获得的人类评估之间的协议。每个WikiEval实例都需要模型比较两个答案或两个上下文片段。我们计算模型（即

[4] https://huggingface.co/datasets/explodinggradients/WikiEval

[5] 也就是说，超出我们在实验中使用模型的报告训练截止点。

	信仰。	答。相关性	续。关系。
Ragas	**0.95**	**0.78**	**0.70**
GPT Score	0.72	0.52	0.63
GPT Ranking	0.54	0.40	0.52

表 1: 在使用 WikEval 数据集的成对比较中，与人类注释者在忠实度、答案相关性和上下文相关性方面的协议（准确性）。

具有最高估计忠实度、答案相关性或上下文相关性的）偏好的答案/上下文与人类标注者偏好的答案/上下文一致的频率。我们以准确率的形式报告结果（即模型与标注者意见一致的实例比例）。

为了将结果置于上下文中，我们将提出的度量标准（表 1 中显示为拉加斯）与两种基线方法进行了比较。对于第一种方法，显示为 *GPT 得分*，我们要求 ChatGPT 为三个质量维度分配一个介于 0 和 10 之间的分数。为此，我们使用了一个描述质量指标意义的提示，然后请求根据该定义对给定的答案/上下文进行评分。例如，在评估忠实度时，我们使用的提示如下：

> 忠实度衡量答案与给定上下文之间的信息一致性。任何在答案中提出的无法从上下文中推断出的主张都应该受到扣分。
> 给定一个答案和上下文，为忠实度分配一个 *0* 到 *10* 范围内的分数。
> 上下文: [上下文]
> 回答: [答案]

平局（即大型语言模型为两个答案候选者分配了相同的分数）通过随机方式打破。第二种基线方法，显示为 *GPT 排名*，则要求 ChatGPT 选择更优的答案/上下文。在这种情况下，提示再次包括所考虑的质量指标的定义。例如，在评估答案相关性时，我们使用的提示如下：

> 答案相关性衡量了回复在多大程度上直接回应并适合给定的问题。它会因问题中出现冗余信息或不完整回答而扣分。给出一个问题和一个答案，根据答案相关性对每个答案进行排序。
> 问题: [问题]
> 答案 *1*: [答案 1]
> 回答 *2*: [答案 2]

表 1 中的结果显示，我们提出的指标与人类的判断非常一致，而两个基线模型的预测则不然。对于忠实度而言，Ragas 的预测总体上高度准确。对于答案相关性，一致性较低，但这主要是因为两个候选答案之间的差异往往十分微妙。我们发现背景相关性是最难评估的质量维度。特别地，我们观察到 ChatGPT 在选择上下文中关键句子的任务上经常遇到困难，特别是在较长的背景下。

6 结论

我们强调了需要对 RAG 系统进行自动化无参考评估的需求。特别是，我们认为有必要建立一个能够评估忠实度（即答案是否基于检索到的上下文）、答案相关性（即答案是否解答了问题）和上下文相关性（即检索到的上下文是否足够集中）的评估框架。为了支持该框架的发展，我们引入了维基评估，这是一个包含人类对这三个不同方面判断的数据集。最后，我们也描述了 Ragas，这是我们实现三个考虑的质量方面的具体方法。这个框架易于使用，并且可以为 RAG 系统的开发者提供有价值的见解，即使在没有地面真实值的情况下也是如此。我们在 WikiEval 上的评估表明，Ragas 的预测与人类预测高度一致，尤其是在忠实度和答案相关性方面。

References

Amos Azaria and Tom M. Mitchell. 2023. The internal state of an LLM knows when its lying. *CoRR*, abs/2304.13734.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman

Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *CoRR*, abs/2302.04166.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. Large language models struggle to learn long-tail knowledge. *CoRR*, abs/2211.08411.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *CoRR*, abs/2212.14024.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *CoRR*, abs/2305.11747.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *CoRR*, abs/2303.08896.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *CoRR*, abs/2305.14251.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *CoRR*, abs/2302.00083.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. REPLUG: retrieval-augmented black-box language models. *CoRR*, abs/2301.12652.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is chatgpt a good NLG evaluator? A preliminary study. *CoRR*, abs/2303.04048.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *CoRR*, abs/2305.17926.

Shufan Wang, Yixiao Song, Andrew Drozdov, Aparna Garimella, Varun Manjunatha, and Mohit Iyyer. 2023c. KNN-LM does not improve open-ended text generation. *CoRR*, abs/2305.14625.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27263–27277.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James R. Glass. 2023. Interpretable unified language checking. *CoRR*, abs/2304.03728.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

A 来自 WikiEval 的示例

表 2、3 和 4 展示了来自 WikiEval 数据集的示例，特别关注高和低忠实度的答案（表 2）、高和低答案相关性（表 3）以及高和低上下文相关性（表 4）。

问题	上下文	回答
Who directed the film Oppenheimer and who stars as J. Robert Oppenheimer in the film?	Oppenheimer is a 2023 biographical thriller film written and directed by Christopher Nolan. Based on the 2005 biography American Prometheus by Kai Bird and Martin J. Sherwin, the film chronicles the life of J. Robert Oppenheimer, a theoretical physicist who was pivotal in developing the first nuclear weapons as part of the Manhattan Project, and thereby ushering in the Atomic Age. Cillian Murphy stars as Oppenheimer, with Emily Blunt as Oppenheimer's wife Katherine "Kitty" Oppenheimer.	高保真度: Christopher Nolan directed the film Oppenheimer. Cillian Murphy stars as J. Robert Oppenheimer in the film. 低保真度詹姆斯·卡梅隆执导了电影《奥本海默》。汤姆·克鲁斯在片中饰演 J. 罗伯特·奥本海默。

表 2: 来自 WikiEval 的示例，展示高保真度和低保真度的答案。

问题	答案
When is the scheduled launch date and time for the PSLV-C56 mission, and where will it be launched from?	高答案相关性: The PSLV-C56 mission is scheduled to be launched on Sunday, 30 July 2023 at 06:30 IST / 01:00 UTC. It will be launched from the Satish Dhawan Space Centre, Sriharikota, Andhra Pradesh, India. 低答案相关性计划发射日期和时间尚未提供。PSLV-C56 任务是印度一项重要的太空任务。它的目标是将一颗卫星送入轨道以研究天气模式。

表 3: 来自 WikiEval 的示例，展示了答案相关性较高和较低的答案。

问题	上下文
When was the Chimnabai Clock Tower completed, and who was it named after?	高语境相关性: The Chimnabai Clock Tower, also known as the Raopura Tower, is a clock tower situated in the Raopura area of Vadodara, Gujarat, India. It was completed in 1896 and named in memory of Chimnabai I (1864 – 1885), a queen and the first wife of Sayajirao Gaekwad III of Baroda State. 低语境相关性奇姆纳拜钟楼，又称拉奥普拉塔楼，位于印度古吉拉特邦瓦多达拉的拉奥普拉地区。该钟楼于 1896 年完工，并以巴罗达州王公赛亚吉劳·盖克瓦德三世的第一位妻子、皇后奇姆纳拜一世（1864 – 1885）的名字命名。它采用了印度-萨拉森建筑风格。历史。奇姆纳拜钟楼建于 1896 年。这座塔以巴罗达州王公赛亚吉劳·盖克瓦德三世的第一位妻子、皇后奇姆纳拜一世（1864 – 1885）的名字命名。它的落成典礼由巴罗达末代纳瓦布米尔·卡马鲁丁·侯赛因汗主持。在盖克瓦德统治时期，这里是马拉车电车的停靠站之一。钟楼的建设费用为 25,000（相当于 2023 年的 920 万或 12 万美元）。

表 4: 来自 WikiEval 的示例，展示具有高上下文相关性和低上下文相关性的答案。