

Q²:Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering (通过问题生成和问答评价知识基础对话中的事实一致性)

Or Honovich Leshem Choshen Roe Aharoni Ella Neeman

Idan Szpektor Omri Abend

The Hebrew University of Jerusalem; Google Research

EMNLP 2021

汇报人：修娜娜

论文工作-背景

- 对话生成模型的输出与依赖的知识在事实上不一致, 甚至完全是幻觉。由于不一致的生成文本通常是流畅且格式良好的, 因此这些输出可能会用错误信息误导用户, 从而限制了此类系统的适用性。
- 对话系统是开放式任务, 不适合机器翻译、抽象摘要等现有参考标准。因为对话输出可能涉及对话历史, 包括个人意见、对用户的问题和一般的“闲聊”, 其与外部知识的一致性在大多数情况下是无关的。

Topic: Asthma

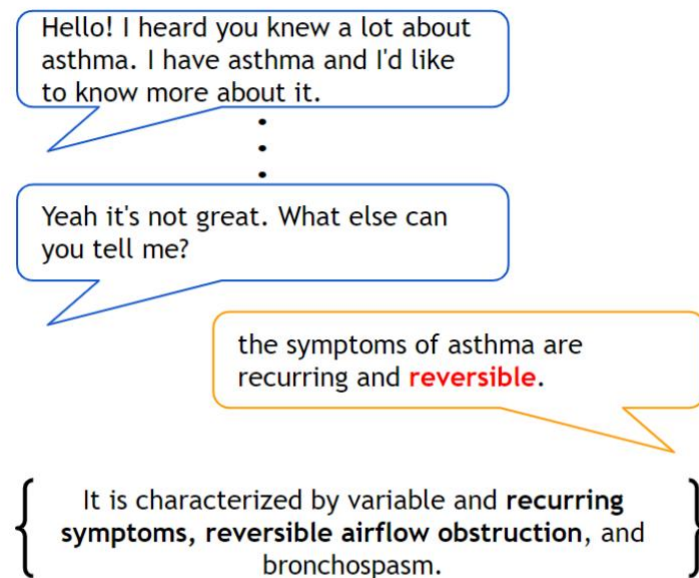
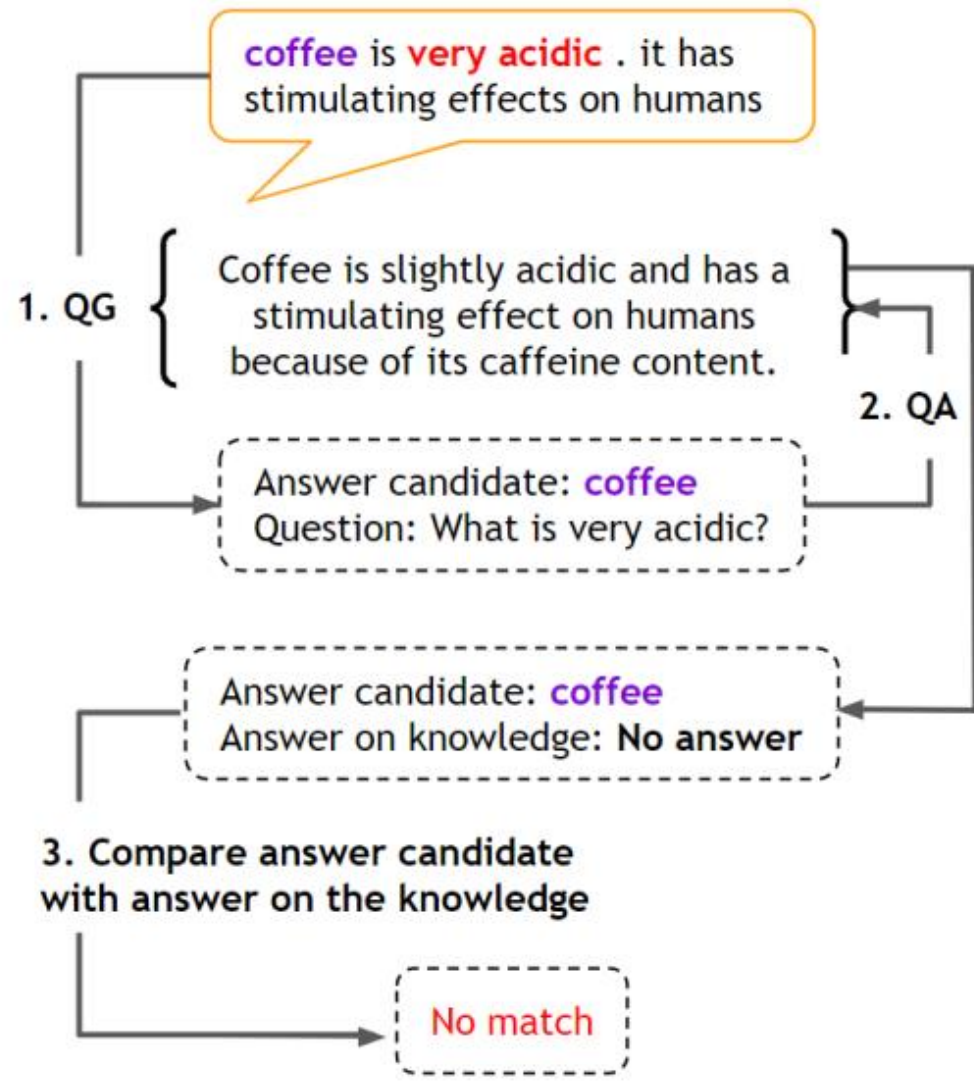


Figure 1: An example from our dataset. Human messages are in Blue, the generated response is in Orange and the grounding knowledge is in Black at the bottom. The factual inconsistency is marked in Red.

>> 论文工作-思路

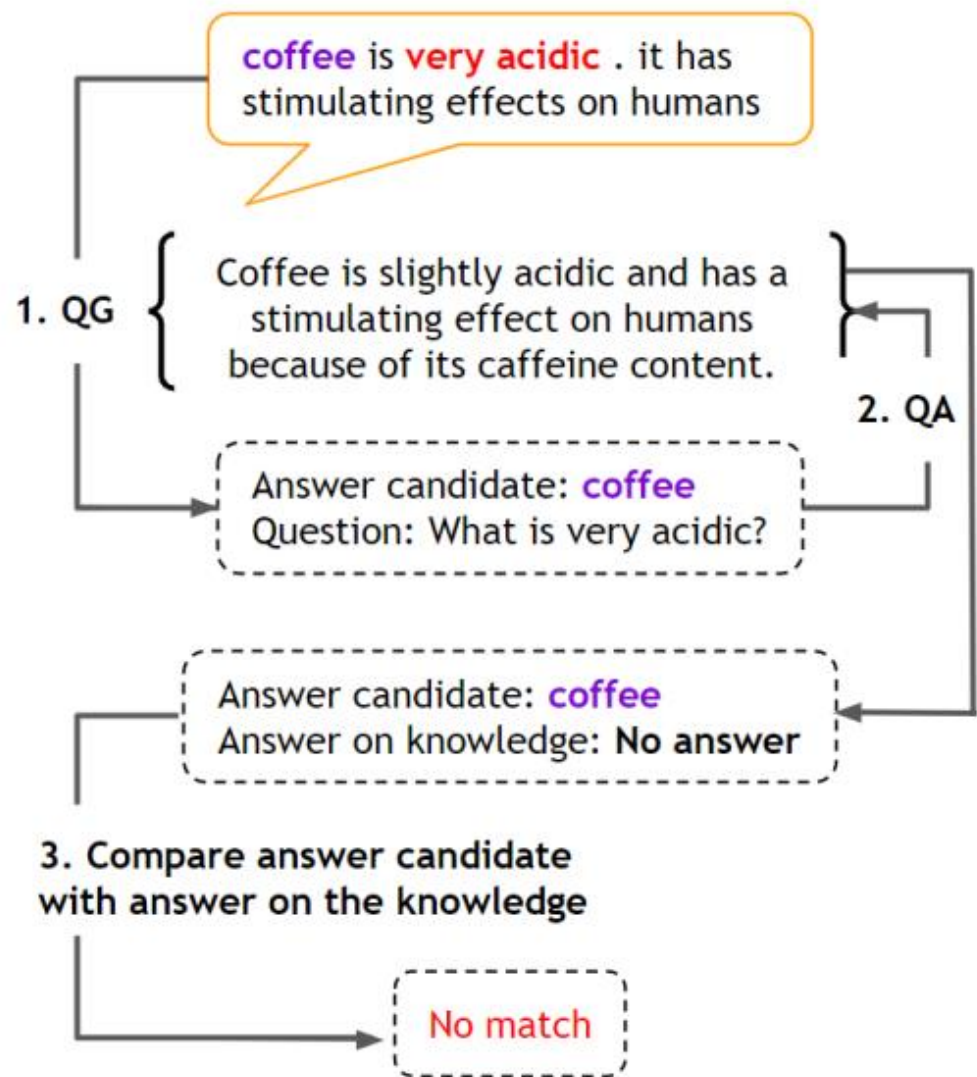
- 提出自动度量 Q^2 ，对自动问题生成（QG）和问题回答（QA）进行生成评估
- 首先将给定生成的响应（回答）作为输入，并使用QG系统生成答案候选（响应中的信息spans）
- 使用QA系统在知识中找到相应的答案spans
- 之前的QG/QA使用标记匹配，本文使用NLI（natural language inference）



论文工作-做法

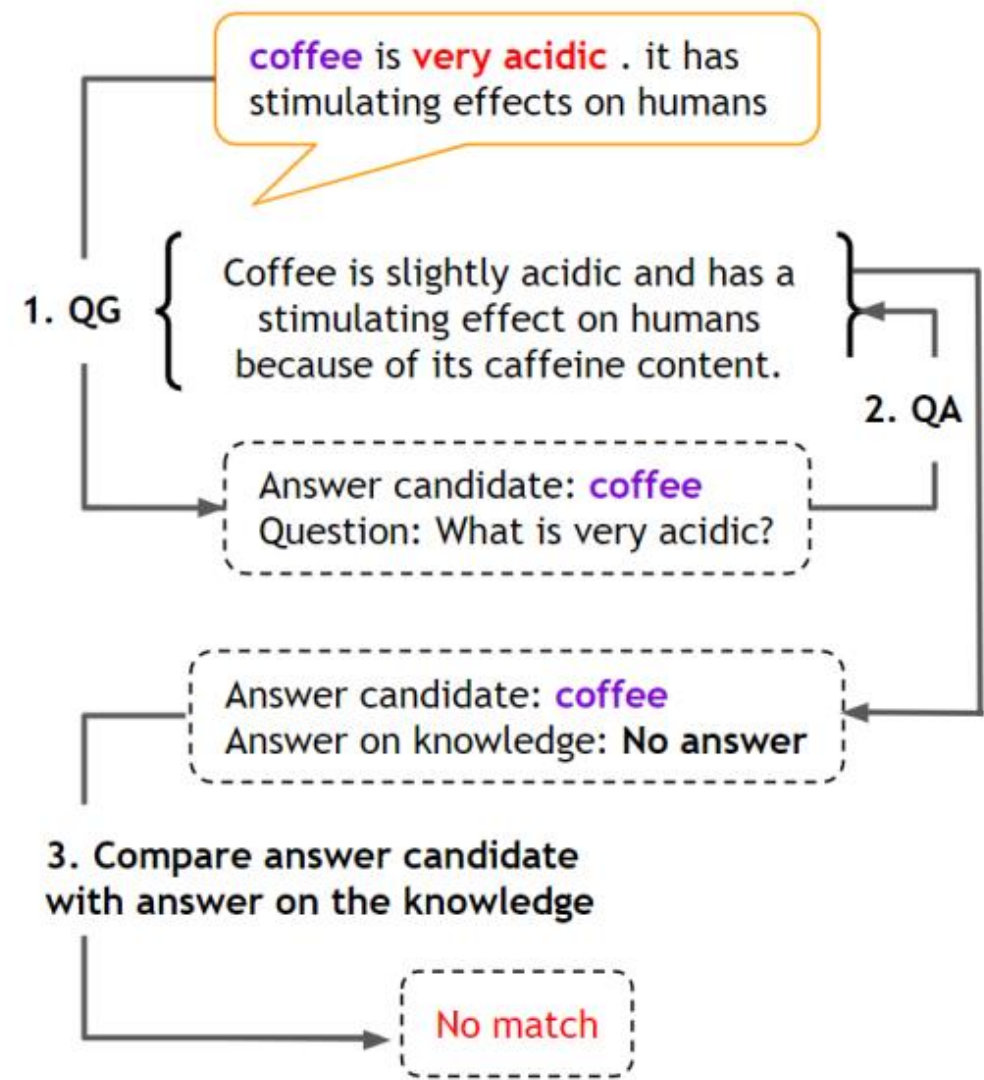
问题生成:

- 在response中标记information span, 作为QG的目标回答。使用spaCy标记r中所有命名实体和名词短语。如左图中“coffee is very acidic”中的“coffee”
- 使用r和information span, 生成问题q, 并以a作为回答。如“coffee”和“What is very acidic?”
- 使用波束搜索解码, 取前5个问题, 报告过滤后排名最高的问题。
- 使用T5-base 微调 SQuAD1.1作为QG模型



问题回答：

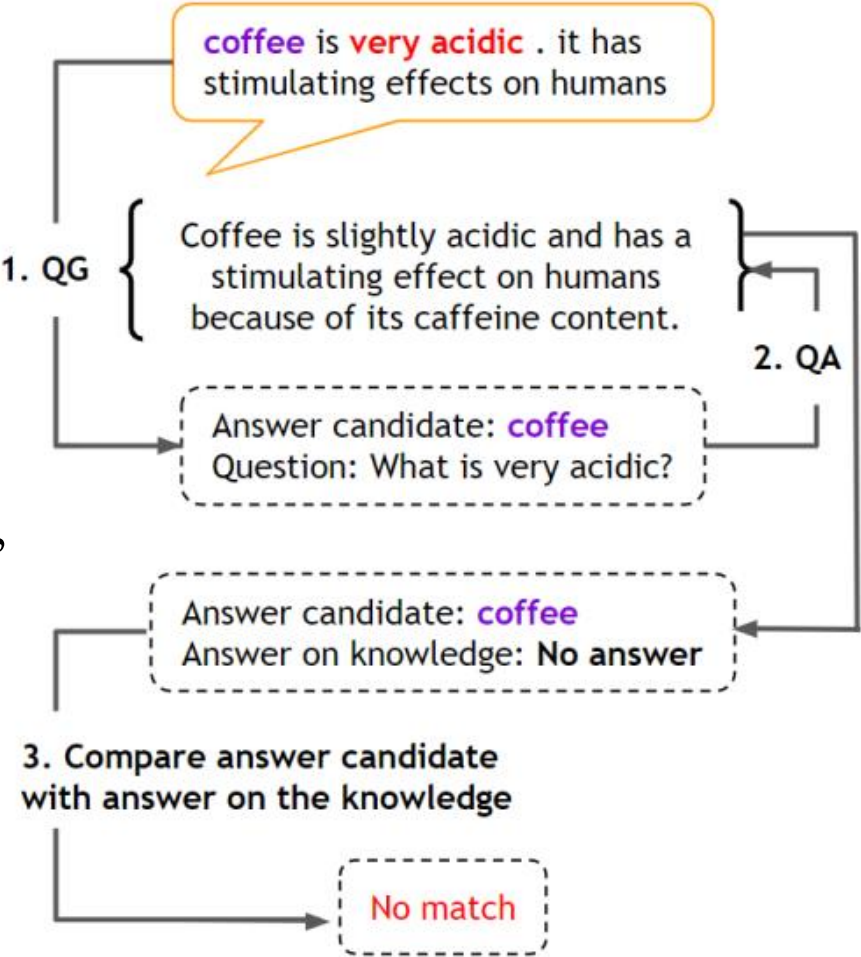
- 使用QA模型回答上述问题：在knowledge找到相应的span
- 使用Albert-Xlarge 微调 SQuAD2.0作为QA模型



论文工作-做法

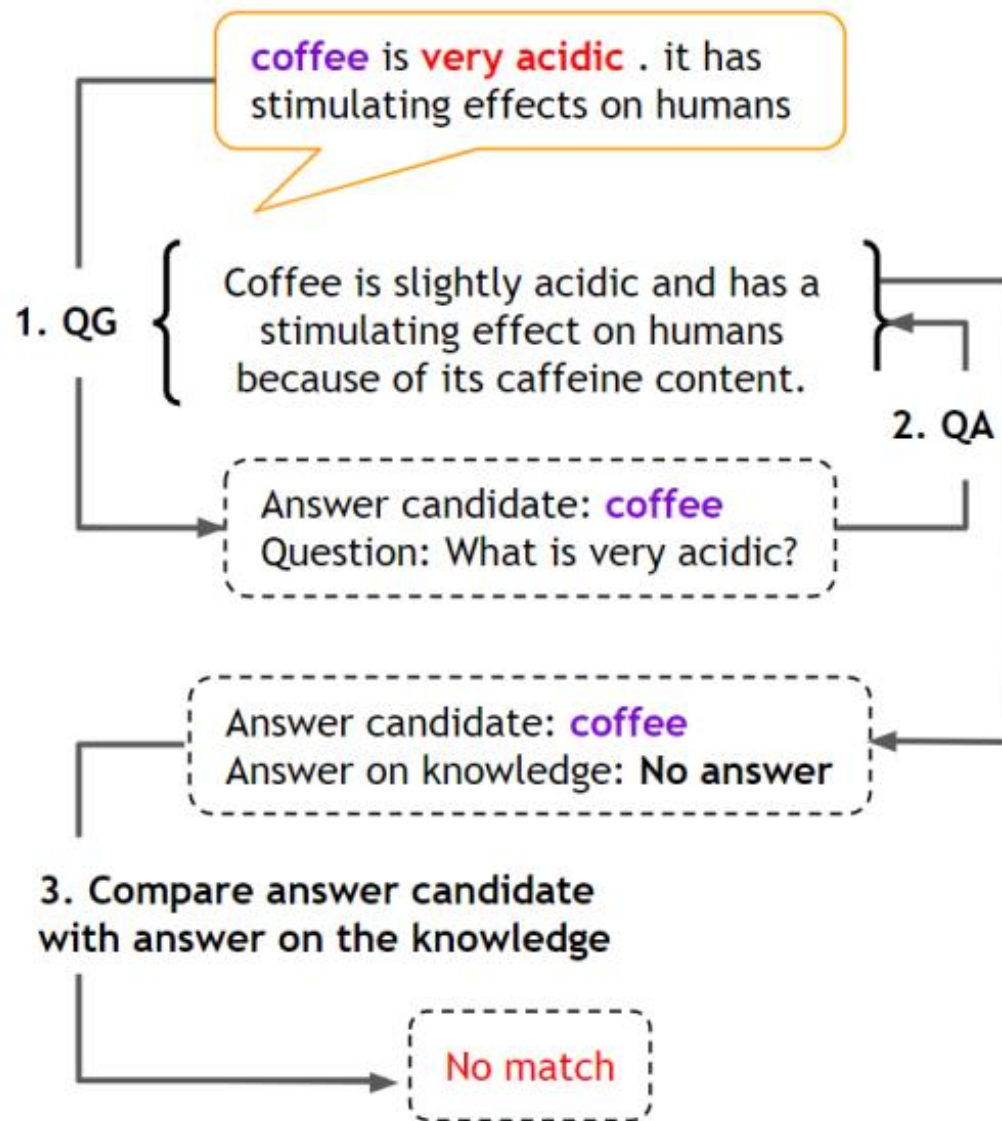
Answer 相似度：

- 为了适应词汇变异，使用NLI模型衡量答案的相似性，如“US” vs. “United States”、 “a book series” vs. “a set of novels”
- 使用knowledge answer为前提， response answers为假设
- 每个答案连接在问题之后，为NLI模型添加上下文
 - question “Where were the Red Hot Chili Peppers formed?”
 - response answer “LA”
 - the knowledge answer “Los Angeles”
 - NLI premise “Where were the Red Hot Chili Peppers formed? Los Angeles”
 - NLI hypothesis “Where were the Red Hot Chili Peppers formed? LA”



分数计算:

- entailment为1, contradiction或无答案为0, neutral使用token level F1
- 最后对所有答案对的匹配分数取平均
- RoBERTa 微调 SNIL作为AllenNLP的实现 (NLI模型)

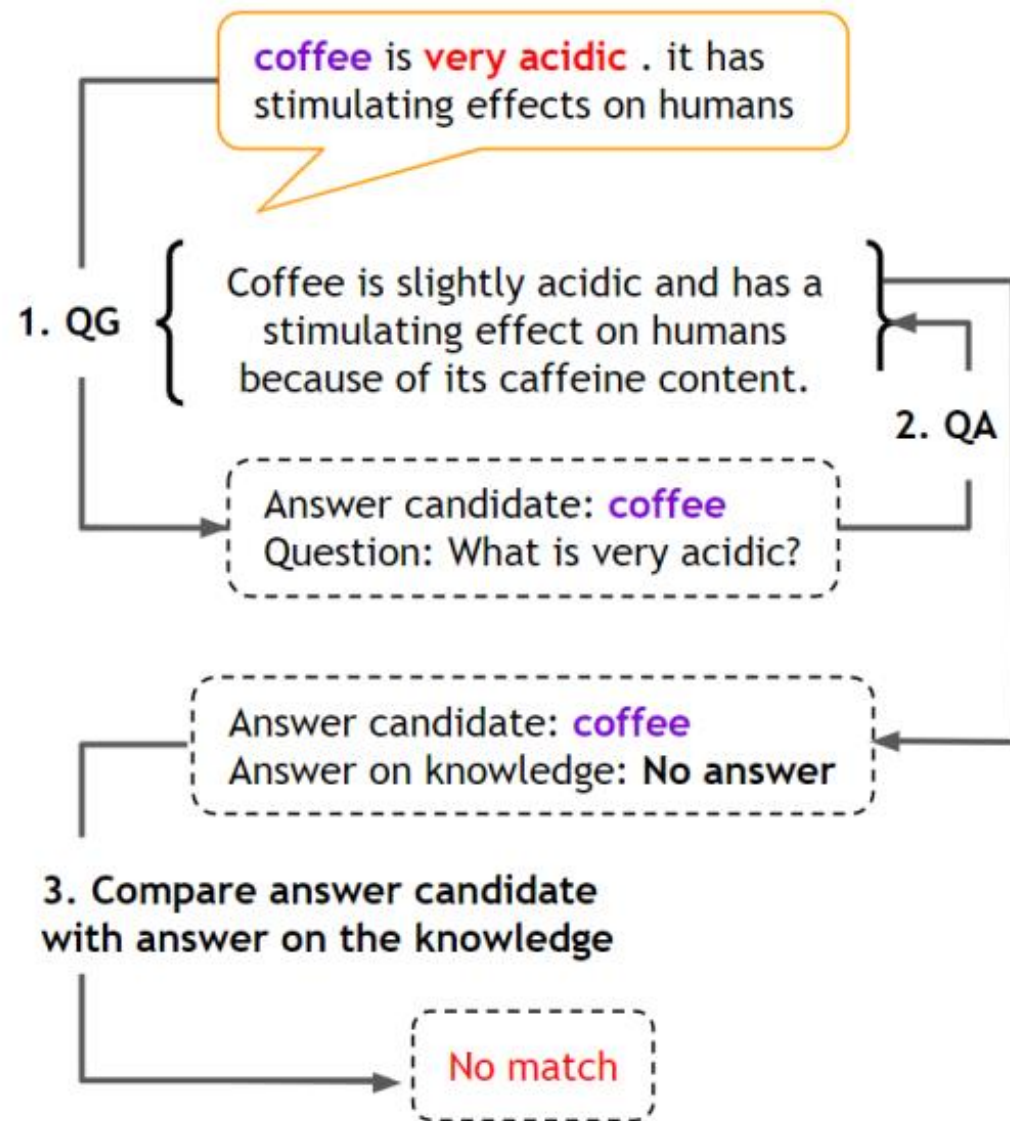


问题筛选

- 运行QA模型，以响应r作为输入段落来回答q，丢弃答案不一致（identical）的问题
- 过滤设计人称代词的问题。评估事实的一致性时希望忽略回应中不符合事实的固执己见的部分。

缺乏有效的问题

- 对于某些响应，没有生成有效的问题，使用NLI模型作为fallback，k为假设，r为前提，entailment为1，contradiction为0，neutral为0.5



评估基准（用于QG、QA模型训练）

- Wizard of Wikipedia（维基百科）数据集
- Topical-Chat数据集
- Dialogue NLI数据集

评价指标：token-level F1、BLUE、BERTScore

重新注释数据集（作为知识K）

- 三位作者做事实一致性注释
- 基于WoW（维基百科）的MenNet、dodecaDialogue的两个对话系统收集response
- 使用波束搜索
- 浏览response，每个系统注释150个事实一致和150个事实不一致的例子，共600个
- 跳过没有基础知识的闲聊

| Topic | Response | Knowledge |
|----------------|--|---|
| Coffee | coffee is very acidic . it has stimulating effects on humans. | Coffee is slightly acidic and has a stimulating effect on humans because of its caffeine content. |
| French cuisine | in that time italian cuisine was influenced by french cuisine | During that time, French cuisine was heavily influenced by Italian cuisine . |
| Madonna | she was born in 1968 and raised in new york city . | Born and raised in Michigan , Madonna moved to New York City in 1978 to pursue a career in modern dance. |
| Sephora | me too! it's an american fashion company founded in 1854 . | Sephora is a French chain of cosmetics stores founded in 1969 . |

Table 1: Examples for factually inconsistent responses from our dataset. Factual inconsistencies are marked in red, with their corresponding parts in the knowledge marked in blue. The first two examples are outputs of the *dodeca*Dialogue system, and the last two are outputs of MemNet.

主实验

- response级评价，Q²更善于自动分离一致和不一致的实例
- 系统级评价，Q²在评估系统层面上的事实一致性更好

分析

- 对底层基础模型变化具有的鲁棒性：将组件模型（QG，QA，NLI）换成小模型，几乎不影响稳健性
- 缺乏有效的问题：对于某些回答，Q²不会产生有效的问题，闲聊和人称指代占有一定比例

案例分析

A: i' m not sure about that but i do know that they are reliant on vulnerable species!

information span: vulnerable species

knowledge: "The giant panda is a conservation reliant vulnerable species. "

Q: What are they reliant on?

answer: conservation

score: 0

- 绘制不同类型response内容，如闲聊、角色和事实，根据其事实来源评估每一种response
- 将Q2应用于事实一致性至关重要的其他任务，例如自动事实检查(Thorne和Vlachos, 2018)，并使用其评估信号来提高Rashkin等人(2021)或Nan等人(2021)提出的生成模型的事实一致性。

Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback (利用外部知识和自动反馈改进大型语言模型)

**Baolin Peng Michel Galley Pengcheng He Hao Cheng Yujia Xie
Yu Hu Qiuyuan Huang Lars Liden Zhou Yu Weizhu Chen Jianfeng Gao**
Microsoft Research Columbia University

汇报人：修娜娜

- 大型语言模型（LLMs）往往会产生幻觉，这在部署到关键任务时可能会造成损害。
- 即使模型规模呈指数级增长，LLMs也无法对许多应用程序所需的所有信息进行编码。大多数以前提出的方法都需要调整LLM的参数，随着LLM的大小呈指数级增长，这可能会非常昂贵。因此，非常需要在固定的LLM中增加即插即用 (PnP) 模块，以执行关键任务。

论文工作-背景

- 本文开发了一个名为LLM-AUGMENTER的系统，可以通过纳入外部知识和迭代修改提示来提高LLM的性能。有了这个系统，LLM可以为各种任务（如问题回答和面向任务的对话）产生更准确和接地气的反应。
- 本文还讨论了攻击性内容生成的潜在危害，并强调了鼓励用户检查补充生成文本的外部来源的相关性的重要性。

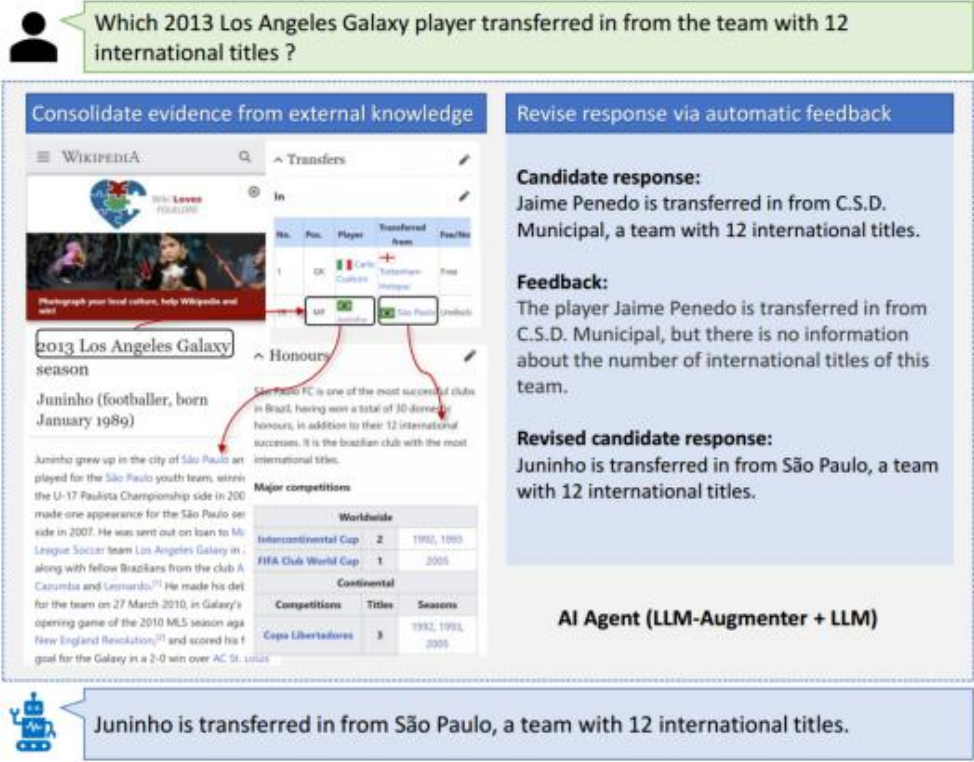


Figure 1: LLM-AUGMENTER improves a fixed LLM by (1) consolidating evidence from external knowledge for the LLM to generate responses grounded in evidence, and (2) revising LLM's (candidate) responses using automated feedback.

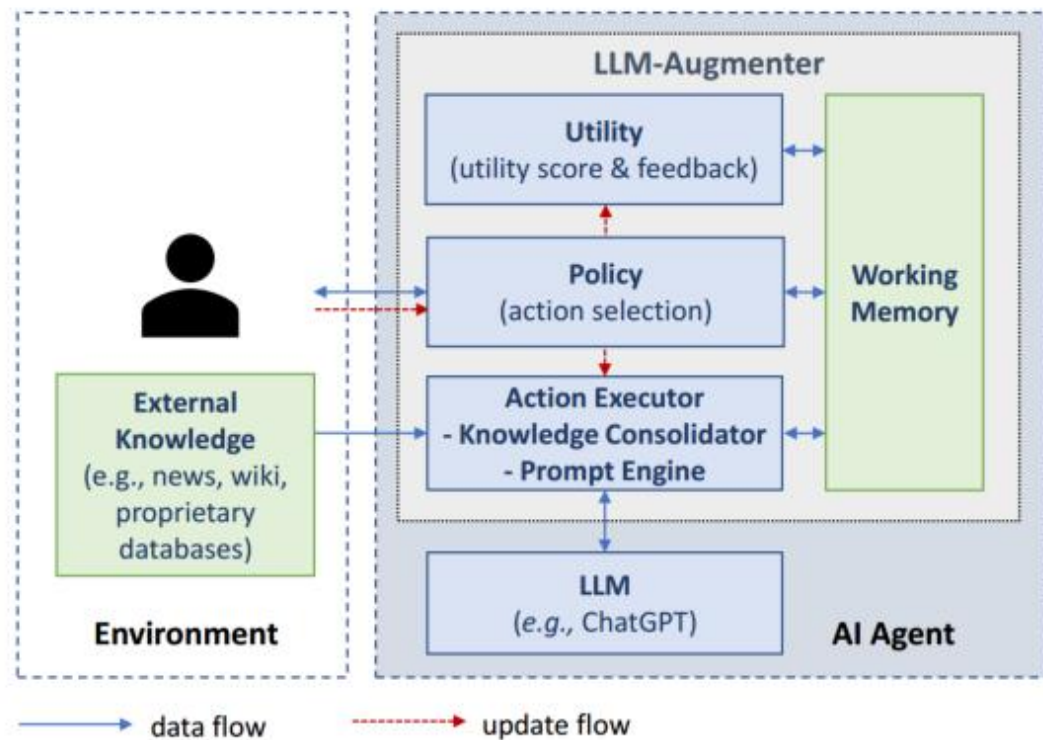


Figure 2: LLM-AUGMENTER architecture showing how its plug-and-play modules interact with the LLM and the user's environment.

- **Working Memory:** 该模块跟踪捕获对话中到目前为止的所有重要信息的对话状态 (q, e, o, u, f, h_q)
- q : query
- e : evidence (Knowledge Consolidator)
- o : response for q
- u : score assessing the utility of o
- f : feedback
- h_q : dialog history before q

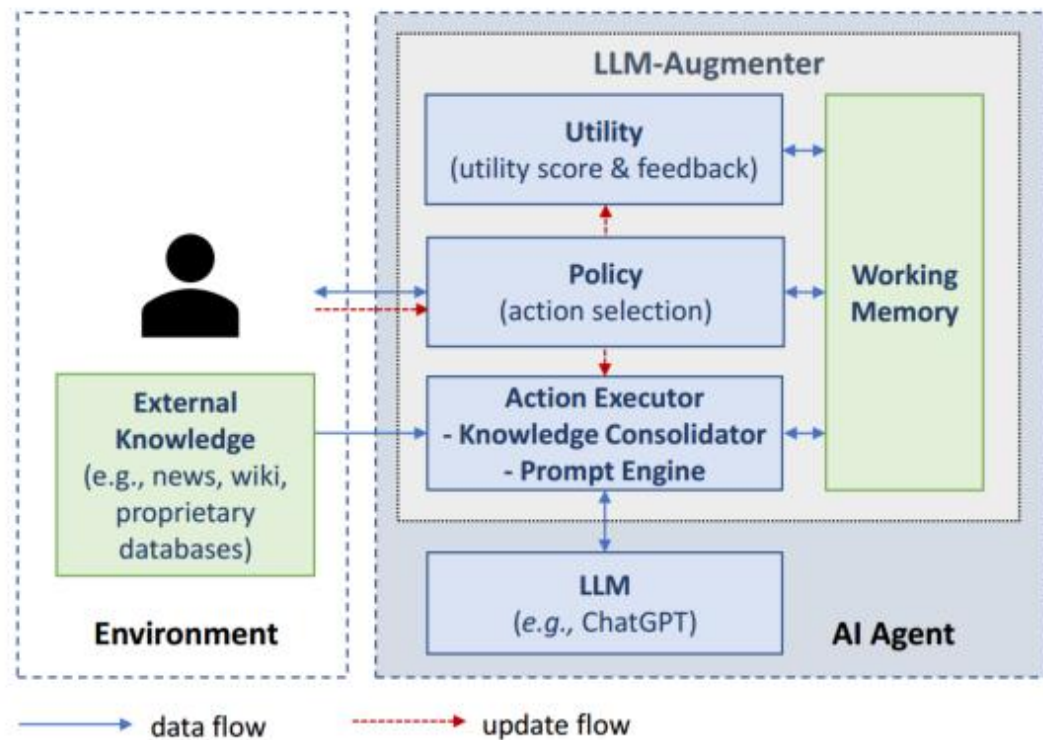


Figure 2: LLM-AUGMENTER architecture showing how its plug-and-play modules interact with the LLM and the user's environment.

- **policy**: 该模块选择下一个导致最佳预期奖励R的系统动作。
 - (1) 从外部知识中获取q的证据e
 - (2) 调用LLM生成候选响应
 - (3) 如果通过了Utility模块的验证，则向用户发送响应

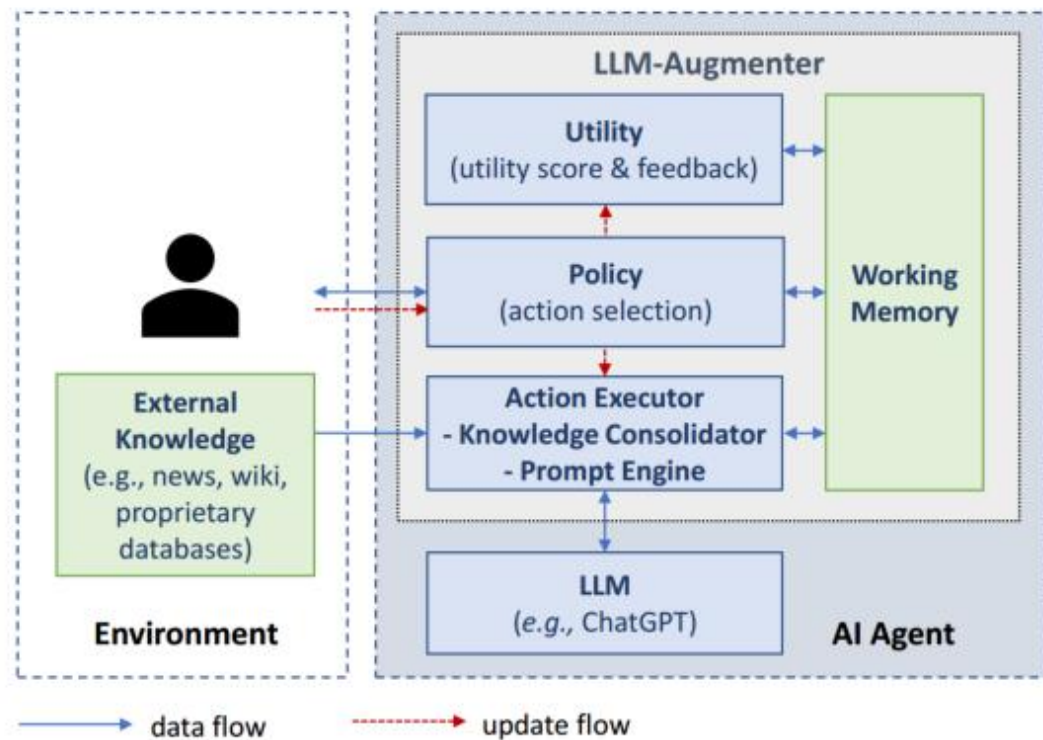


Figure 2: LLM-AUGMENTER architecture showing how its plug-and-play modules interact with the LLM and the user's environment.

- **Action Executor:** 该模块执行policy选择的action。它由**知识整合器**和**提示引擎**两部分组成。
- **Knowledge Consolidator:** 模块化设计，由knowledge retriever, entity linker, evidence chainer组成
- knowledge retriever: 检索原始证据
- entity linker: 丰富原始证据（链接实体和相应描述）
- evidence chainer: 修剪不相关的证据

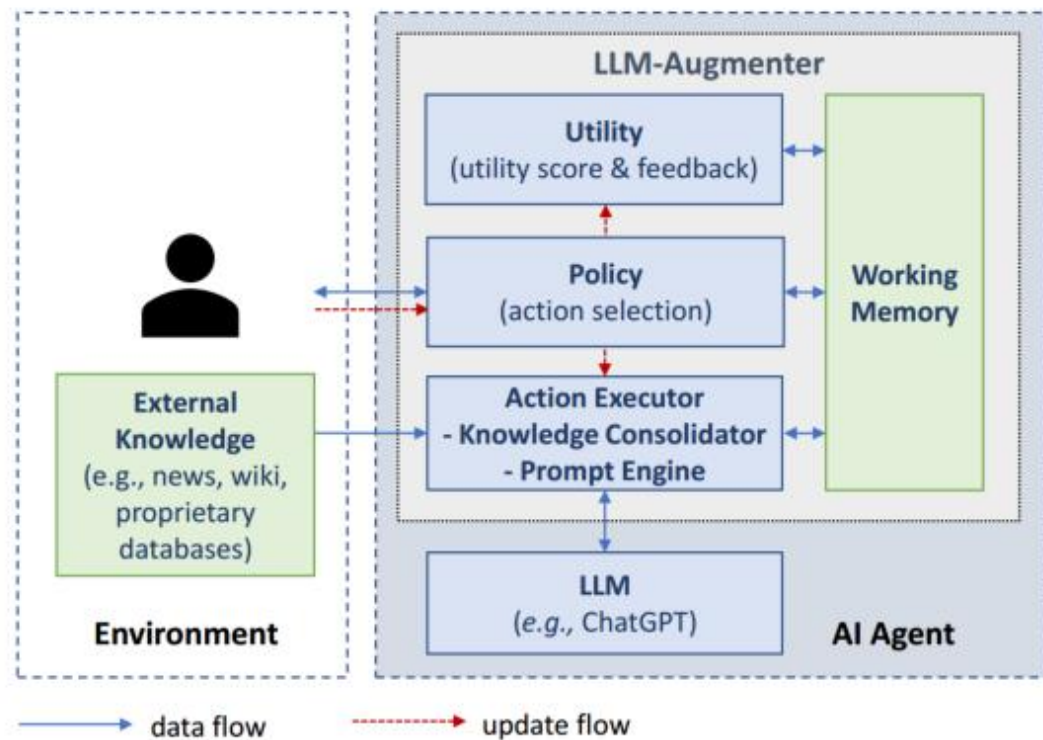


Figure 2: LLM-AUGMENTER architecture showing how its plug-and-play modules interact with the LLM and the user's environment.

- **Prompt Engine:** 生成一个查询LLM的prompt，为q生成一个(候选)响应o。提示是一个文本字符串，由任务指令、用户查询q、对话历史q、证据e(如果它被知识整合器提供)和反馈(如果它被实用程序模块提供)组成。提示是特定于任务的。

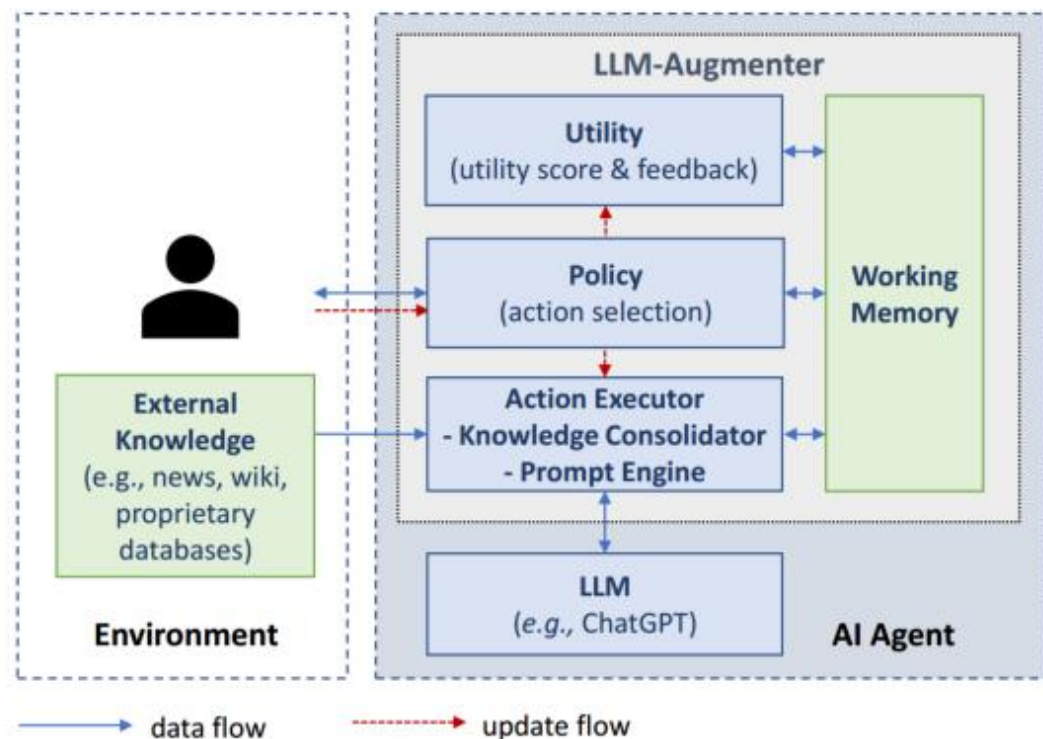


Figure 2: LLM-AUGMENTER architecture showing how its plug-and-play modules interact with the LLM and the user's environment.

- **Utility:** 给定候选响应 o , Utility模块使用一组特定于任务的utility函数生成utility得分 u 和相应的反馈 f 。
- Model-based utility functions: 回答是否符合人类偏好
- Rule-based utility functions: 回答是否符合特定规则
- f : 生成信息丰富且可操作的反馈, 以帮助修改prompt, 使LLM能够生成更好的响应。

右图对比ChatGPT和LLM-AUGMENTER的示例响应。

- ChatGPT无法提供与用户相关的特定知识相关的响应，例如，当地的印度餐馆。
- LLM-augmenter的工作记忆从外部知识中检索到的更丰富的信息，以帮助底层LLM(即ChatGPT)生成更有内容的响应。
- LLM-augmenter收到的第一个LLM响应并不令人满意，因为LLM生成的质量和特殊性是不可预测的。
- Utility模块确定第一个响应不符合其标准(即，KF1高于给定阈值)，并向LLM模块发出反馈(即，“响应与知识不一致”)。根据效用函数，LLM-AUGMENTER收到的第二个响应更令人满意，因此发送给用户。

| | |
|------------------------|--|
| Instruction | I want you to act as a chatbot AI for travel planning. You need to answer customer's questions nicely. |
| Context | User: Hello, I am looking for a moderately priced restaurant on the north side of town. Assistant: I recommend golden wok. It is in the north part of town with a moderate price range. User: Yes, I am looking for a moderately priced restaurant in the north part of town. Assistant: I recommend golden wok. It is in the north part of town with a moderate price range. User: I don't want Golden Wok, is there anything else?, Assistant: Yes, your other option is The Nirala. It serves Indian food and is also located in the north part of town. User: Is their beer a good value for the money? |
| ChatGPT Response | I am sorry, I do not have information on the beer prices at The Nirala. It is best to contact the restaurant directly for more information. |
| Working Memory | Review: They also have a modest drinks menu with some affordable prices, but I was more interested in the beautiful view from my window seat, which allowed me to watch the sun setting as I dined. Q: Is alcohol served at this restaurant? A: Yes, alcohol is served at this restaurant. Review: I found them to be overpriced and mediocre. Review: The drinks ae priced well. Review: Something that I disliked, however was the prices for their drinks. |
| LLM-AUGMENTER Response | I am sorry, I do not have information on the value of their beer. I only have information on the location and price range of the restaurant. |
| Rule-based Feedback | The response is inconsistent with the knowledge. Please generate again? |
| LLM-AUGMENTER Response | Some reviews mention that the drinks at the restaurant are priced well, while others mention that they found the prices to be overpriced and mediocre. |
| Human Response | Unfortunately the reviews are split, half say they are overpriced and have say they are affordable. |

Table 6: Prompt and example responses of different models for the customer service scenarios.

- 验证LLM-AUGMENTER在两种不同场景下的有效性：
 - (1) **信息搜索对话**，AI代理需要基于各种外部知识来源生成信息丰富且值得信赖的响应；
 - (2) **Wiki问答**，AI代理需要通过拼凑分散在多个Wiki文档中的各种模式的信息来回答问题。

• 信息搜索对话

results:

- 由于外部知识和自动反馈，LLM-AUGMENTER减轻了ChatGPT在新闻聊天和客户服务任务上的幻觉问题。

| Model | K.C. | Feedback | KF1 ↑ | BLEU ↑ | ROUGE ↑ | chrF ↑ | METEOR ↑ | BERTScore ↑ | BARTScore ↑ | BLEURT ↑ | Avg. length |
|---------------|------|----------|--------------|-------------|--------------|--------------|--------------|--------------|-------------|--------------|-------------|
| CHATGPT | - | - | 26.71 | 1.01 | 16.78 | 23.80 | 7.34 | 82.14 | 0.25 | 26.98 | 58.94 |
| LLM-AUGMENTER | BM25 | ✗ | 34.96 | 6.71 | 22.25 | 27.02 | 9.35 | 83.46 | 0.34 | 26.89 | 46.74 |
| LLM-AUGMENTER | BM25 | ✓ | 36.41 | 7.63 | 22.80 | 28.66 | 10.17 | 83.33 | 0.35 | 27.71 | 54.24 |
| LLM-AUGMENTER | gold | ✗ | 57.44 | 19.24 | 38.89 | 40.02 | 17.21 | 86.65 | 0.82 | 40.55 | 44.35 |
| LLM-AUGMENTER | gold | ✓ | 60.76 | 21.49 | 40.56 | 42.14 | 18.50 | 86.89 | 0.93 | 42.15 | 47.19 |

Table 1: Evaluation scores (in %) and average response lengths for the News Chat (DSTC7) dataset. BM25: Each model retrieves 5 knowledge snippets from the corresponding knowledge source. K.C. denotes Knowledge Consolidator.

| Model | K.C. | Feedback | KF1 ↑ | BLEU ↑ | ROUGE ↑ | chrF ↑ | METEOR ↑ | BERTScore ↑ | BARTScore ↑ | BLEURT ↑ | Avg. length |
|---------------|------|----------|--------------|-------------|--------------|--------------|--------------|--------------|-------------|--------------|-------------|
| CHATGPT | - | - | 31.33 | 4.70 | 24.02 | 27.14 | 12.83 | 87.88 | 1.53 | 47.99 | 28.81 |
| LLM-AUGMENTER | BM25 | ✗ | 34.07 | 4.78 | 24.52 | 28.95 | 13.61 | 87.96 | 1.78 | 47.21 | 32.65 |
| LLM-AUGMENTER | BM25 | ✓ | 37.41 | 3.86 | 24.20 | 30.90 | 14.74 | 87.58 | 2.09 | 44.71 | 45.07 |
| LLM-AUGMENTER | gold | ✗ | 45.63 | 6.54 | 29.77 | 33.32 | 16.93 | 89.35 | 2.59 | 54.38 | 33.04 |
| LLM-AUGMENTER | gold | ✓ | 52.83 | 5.63 | 29.65 | 35.68 | 18.66 | 89.01 | 3.14 | 52.49 | 45.09 |

Table 2: Evaluation scores (in %) and average response lengths for the Customer Service (DSTC11) dataset. BM25: Each model retrieves 5 knowledge snippets from the corresponding knowledge source. K.C. denotes Knowledge Consolidator.

- 信息搜索对话
- The impact of using trainable Policy
- LLMAUGMENTER对测试数据的奖励随着训练集(对话会话)的增加而增加, 在600次交互后超过随机策略, 最终达到约37.5的KF1分数。通过这些互动, LLM-AUGMENTER能够学习有效地选择下一个系统动作以最大化奖励, 这有助于系统减少幻觉, 同时产生流畅和信息丰富的反应。

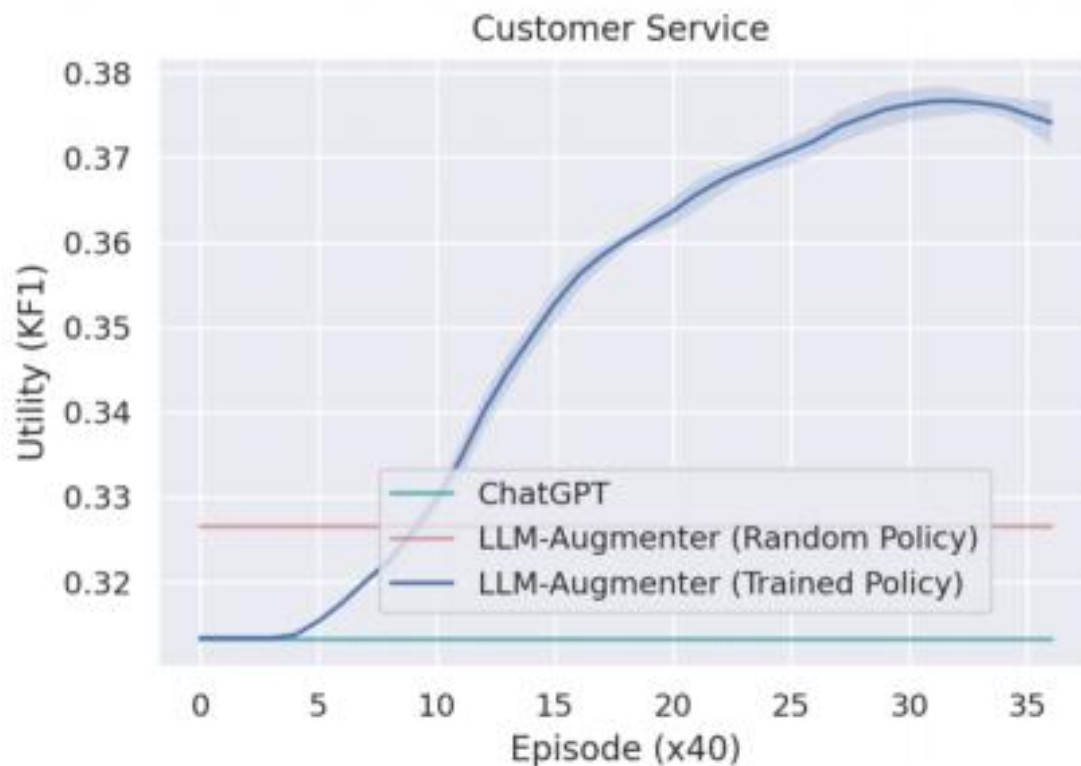


Figure 3: Learning curve of ChatGPT-Companion with T5-Base as the policy module. The solid curves are the mean and the shaded regions are the maximum and minimum utility scores over 5 runs.

- 信息搜索对话

Ablation:

- no-knowledge consolidator
- Self-ask, 知识整合器只有在LM建议使用外部知识时才会被使用, 并提示其是否使用;
- Always-use

results:

Self-ask: 24%的request

Always-use: 获得最佳KF1分数的同时, 也产生了额外的开销。

因此, 应该采用可训练的policy模型来学习何时使用外部知识。

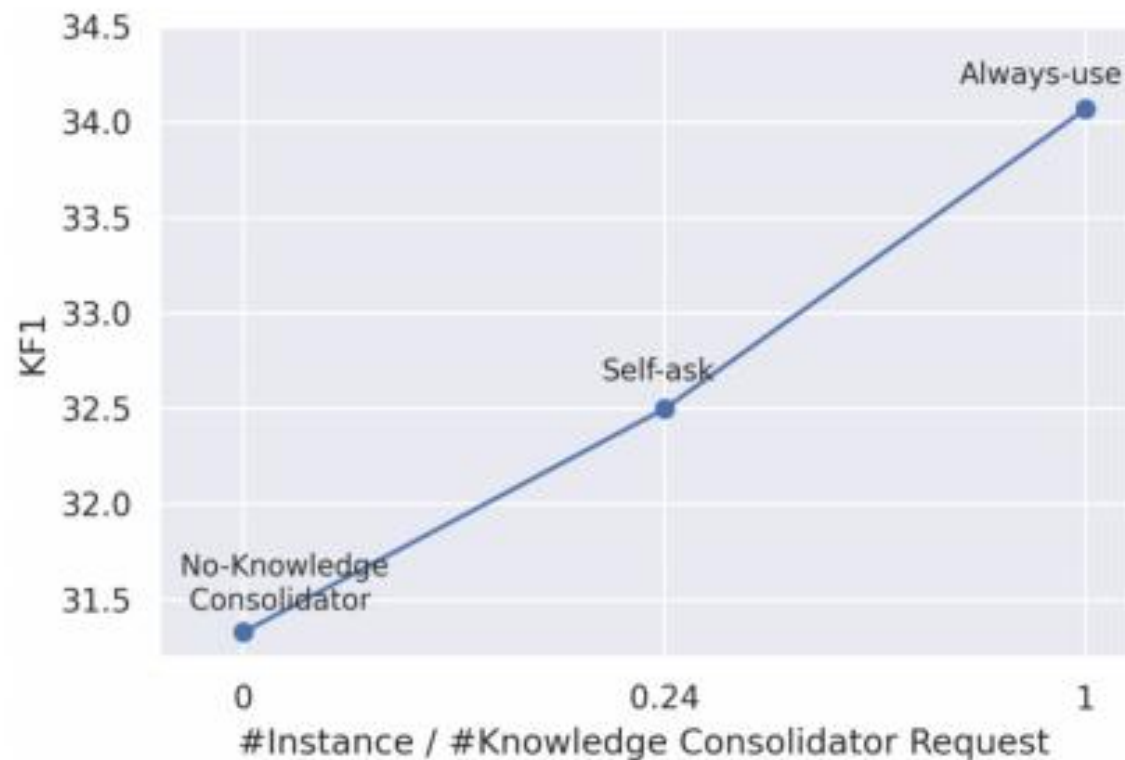


Figure 4: Ablation studies on different policies of LLM-AUGMENTER in Customer Service scenario.

• 信息搜索对话

Ablation:

| Feedback | KF1 \uparrow |
|----------------------------|----------------|
| w/o Feedback | 34.07 |
| w/ Rule-based Feedback | 37.41 |
| w/ Self-criticism Feedback | 37.10 |

Table 4: Ablation studies on different feedback of LLM-AUGMENTER in Customer Service scenario.

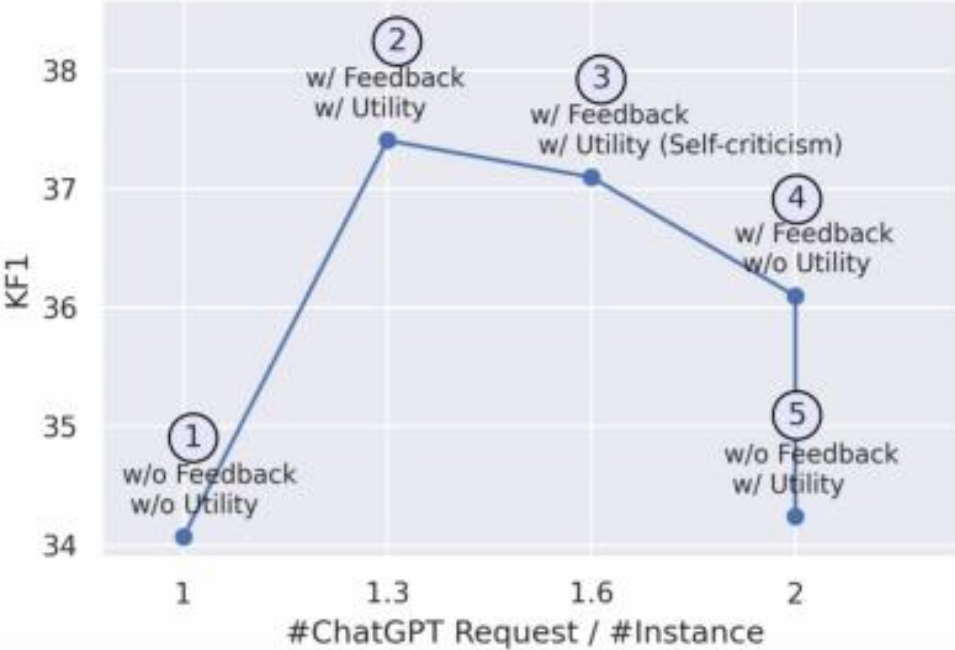


Figure 5: LLM-AUGMENTER benefits from the combination of using utility function and iterative improvement with feedback. The x-axis indicates the average number of ChatGPT prompting and the y-axis is the KF1. The studies are conducted in the Customer Service scenario with knowledge being provided by BM25.

Wiki问答：

与使用前50名综合evidence的最先进的微调模型相比，在性能上仍存在明显差距。

| Model | Knowledge Consolidator | Feedback | Wiki QA | | |
|---------------|------------------------|----------|---------|-------|-------|
| | | | P ↑ | R ↑ | F1 ↑ |
| CHATGPT | - | - | 0.48 | 1.52 | 0.59 |
| LLM-AUGMENTER | DPR | ✗ | 2.08 | 4.31 | 2.38 |
| LLM-AUGMENTER | CORE | ✗ | 7.06 | 14.77 | 8.08 |
| LLM-AUGMENTER | CORE | ✓ | 8.93 | 33.87 | 11.80 |

Table 5: Evaluation results on Wiki QA. Each model retrieves top-5 knowledge snippets from the corresponding knowledge source. The top-5 answer recall of consolidated evidence (CORE) is 50.83.

- 本文的结论是，LLM-AUGMENTER是一个有效的框架，可以用外部知识和自动反馈来增强黑盒LLM。作为LLM提示的一部分提供的外部知识有助于产生更多的反应，这些反应更多的是基于与当前对话相关的外部知识。自动反馈激发了诸如ChatGPT和Instruct-GPT等模型的 "后续修正 "能力，以产生根据一些给定的效用函数排名较高的修正回答。
- 在未来的工作中，作者计划利用与真实用户的互动和用户反馈来训练LLM-AUGMENTER。他们还计划探索使用与冒犯性有关的效用函数，以帮助减轻由冒犯性内容生成造成的伤害。此外，他们强调了鼓励用户检查补充生成文本的外部来源的相关性的重要性，因为外部来源可能会损害生成文本的事实性。最后，他们建议需要进一步研究在互动环境中使用ChatGPT等计算成本高的模型时，如何最好地平衡准确性和效率。

Active Prompting with Chain-of-Thought for Large Language Models

(大语言模型的主动提示与思维链)

Shizhe Diao Pengcheng Wang Yong Lin Tong Zhang

The Hong Kong University of Science and Technology

University of Toronto

汇报人：修娜娜

COT(chain of thought) prompting: 依赖于人类工程学，它需要人类选择一些信息丰富的问题，然后用CoT和答案对它们进行注释。人工注释的范例(带有注释的CoT和答案的问题)对于不同的任务不一定是最有效的。

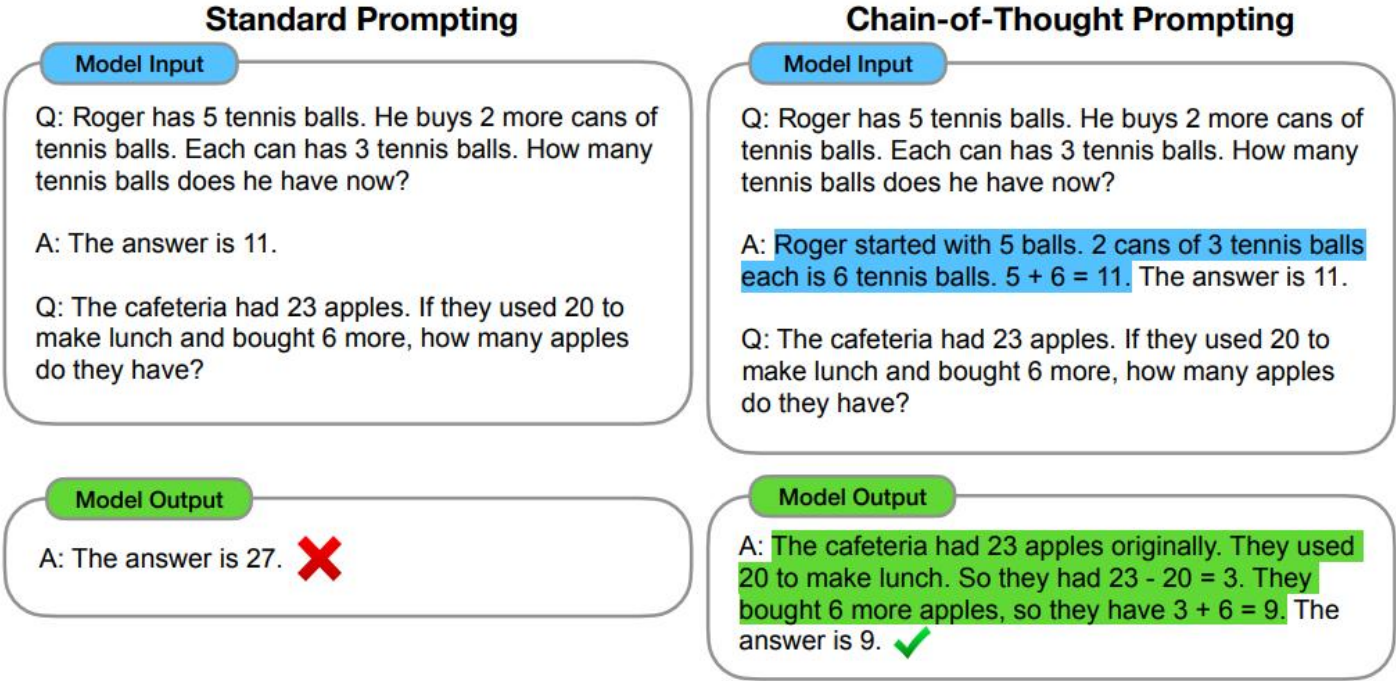


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

例如，原始的思维链提示为八个问题制作了示例，这些示例要么是从训练集中随机选择的，要么是由人类手动组成的。因为推理任务的性质在难度、范围、领域等方面存在着明显的差异，我们不知道什么样的问题最值得标注。同样不清楚的是，一组特定的范例是否能最好地引出所需的信息。

鉴于此得出关键问题，即如何确定哪些问题对注释最重要和最有帮助。前人的研究(Gentile et al, 2022)表明，减少模型的不确定性有助于提高模型的性能。因此，本文引入LLM的不确定性作为选择数据的度量。通过利用**不确定性(uncertainty)**并引入一些人工工作来注释一小组问题。并为每个问题引入几个度量指标来定量描述这种不确定性。

具体：给定数据集 D ，首先要求模型回答 k 次。然后根据每个问题的 k 个答案计算该模型的不确定性 u 。选择最不确定的 n 个 u 最大的问题，并对这些问题进行注释以制作新的范例 E 。最后，按照思维链提示的标准格式，在每个测试问题前添加 E 。（active:主动选择问题）

论文工作-做法

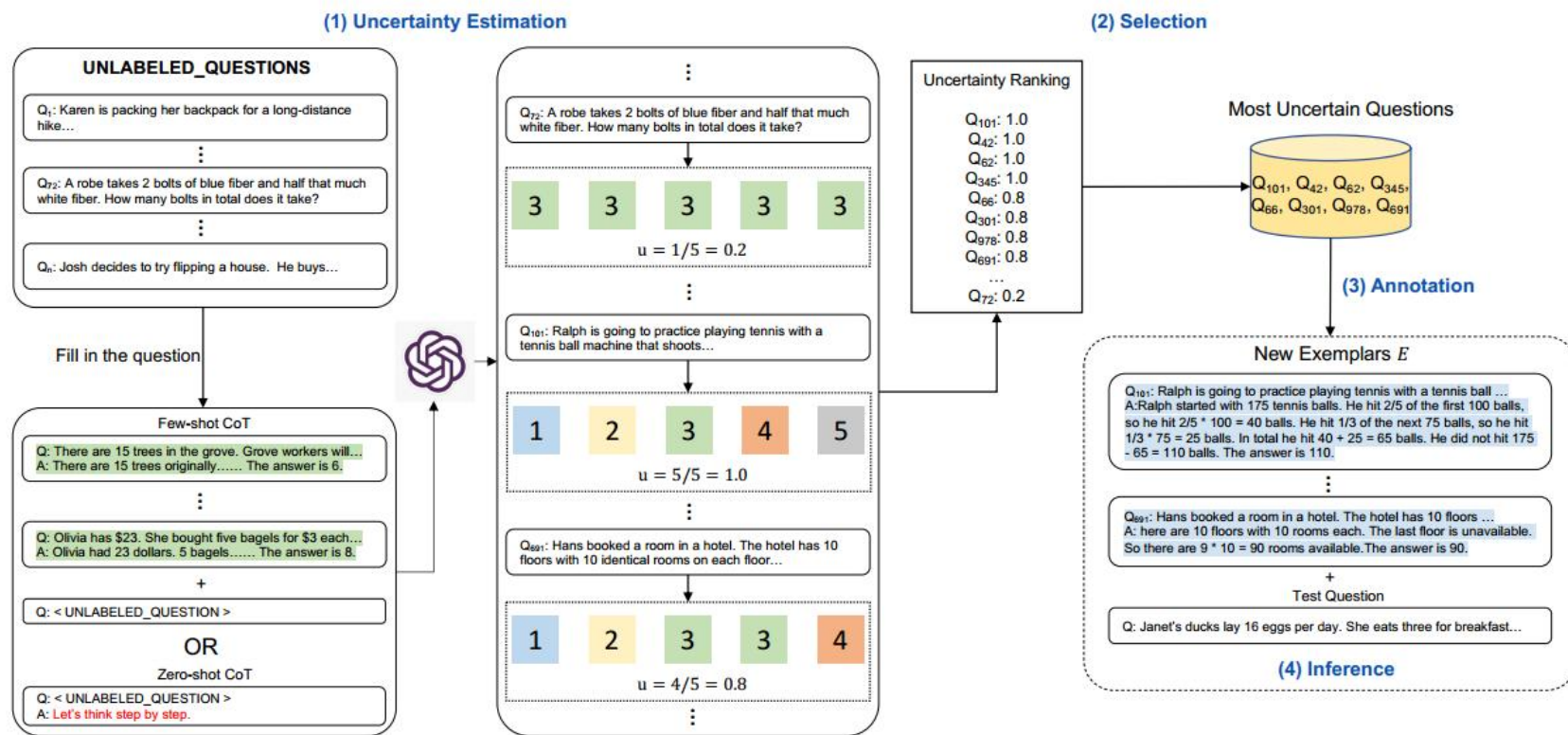


Figure 1: Illustrations of our proposed approach. Four stages: **(1) Uncertainty Estimation:** with or without a few human-written chain-of-thoughts, we query the large language model k ($k = 5$ in this illustration) times to generate possible answers with intermediate steps for a set of training questions. Then we calculate the uncertainty u based on the k answers via an uncertainty metric (we use disagreement in this illustration). **(2) Selection:** according to the uncertainty, we select the most uncertain questions for annotation. **(3) Annotation:** we involve humans to annotate the selected questions. **(4) Inference:** infer each question with the new annotated exemplars.

- Uncertainty Estimation

在COT设置中，首先forward LLM k 次，每个问题得到 k 个答案。然后，一个问题的不确定性可以用不同的方式来衡量。本文考虑四种潜在的不确定性度量：

- Disagreement: disagreement在于计算预测中唯一的答案。实现：我们首先对 k 个生成的答案 $A=\{a_1, a_2, \dots, a_k\}$ 进行集合运算，去除重复项，得到 h 个唯一项 $A=\{a_1, a_2, \dots, a_h\}$ 。disagreement $u=h/k$

- Entropy:
$$u = \arg \max_i - \sum_{j=1}^k P_{\theta}(a_j|q_i) \ln P_{\theta}(a_j|q_i), \quad (1)$$

P_{θ} 表示在所有预测中某一预测结果出现的频率。熵越大表示系统的不确定性越大，熵越小表示不确定性越小。因此，在复杂推理中，将选择熵相对较大的问题作为候选问题。

- Uncertainty Estimation

在COT设置中，首先forward LLM k次，每个问题得到k个答案。然后，一个问题的不确定性可以用不同的方式来衡量。本文考虑四种潜在的不确定性度量：

- Variance:
$$u = \arg \max_i \frac{\sum_{j=1}^k (a_j - \bar{a})^2}{k - 1} \Big|_{q=q_i}, \quad (2) \quad \bar{a} = \frac{1}{k} \sum_{j=1}^k a_j$$

为了缓解大数的支配问题，本文提出对问题中所有提到的数字进行规范化预测：例如，给定一个问题，There are x1 people. Each persc($\hat{y}/(|x_1|+|x_2|)$)? apples. How many apples are there altogether?预测答案y，规范化

- Uncertainty Estimation

在COT设置中，首先forward LLM k次，每个问题得到k个答案。然后，一个问题的不确定性可以用不同的方式来衡量。本文考虑四种潜在的不确定性度量：

- Self-confidence: 对不确定性的估计也可以由LLMs自己来完成，即self-confidence。它可以通过使用手动制作的模板T查询模型来获得，例如对于问题q和预测答案a, report the confidence about the answer from choices.
(a) very confident (b) confident (c) not confident (d) wrong answer. 然后通过以下方式选择最不自信的问题：

$$\begin{aligned} u &= \arg \max_i (1 - \max_j P_{\theta}(a_j | q_i)) \\ &= \arg \min_i \max_j P_{\theta}(a_j | q_i), \end{aligned} \tag{3}$$

首先进行了一项试点研究，发现基于disagreement、entropy和variance的指标在竞争中表现良好，明显优于self-confidence。因此，在本文实验中主要使用disagreement、entropy来实现本文的方法。

- Selection and Annotation

在获得每个问题的不确定度后，可以根据每个问题的不确定度建立不确定度排序。然后我们将选择前 n 个不确定问题进行注释。如果不确定性最大的问题超过 n 个，将从中随机选择 n 个问题。这些问题将由人类注释者通过用逻辑链和答案构建新的范例 E 。 E 将取代最初的 E ，将使用它来进行few-shot COT prompting。

Datasets and Evaluation Metrics

- Arithmetic Reasoning: GSM8K , ASDiv , SVAMP, AQuA, SingleEq .
- Commonsense Reasoning: CSQA, StrategyQA.
- Symbolic:Letter (4)

- SC: CoT的改进版本。它不是贪婪解码，而是对一组推理路径进行采样，并选择最常见的答案。
 - Auto-CoT: 一种基于聚类和zero-shot提示生成基本原理的范例自动构建方法。
 - Random-CoT: Active-Prompt的基线。与ActivePrompt共享相同的注释过程。唯一的区别是，它从训练数据中随机抽取问题进行注释，而不是应用不确定性指标进行主动选择。
 - 基于disagreement (ActivePrompt (D))
 - 基于entropy (Active-Prompt (E))
- 总的来说，本文的模型在很大程度上优于所有基线模型。

| METHOD | GSM8K | ASDiV | SVAMP | AQUA | SINGLEEQ | CSQA | STRATEGY | LETTER (4) | AVG. |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------|-------------|
| Prior Best | 55.0 ^a | 75.3 ^b | 57.4 ^c | 37.9 ^d | 32.5 ^e | 91.2 ^f | 73.9 ^g | - | - |
| UL2-20B | | | | | | | | | |
| CoT | 4.4 | 16.9 | 12.5 | - | - | 51.4 | 53.3 | 0.0 | - |
| SC | 7.3 | 21.5 | 19.4 | 26.9 | - | 55.7 | 54.9 | 0.0 | - |
| LaMDA-137B | | | | | | | | | |
| CoT | 14.3 | 46.6 | 37.5 | - | - | 57.9 | 65.4 | 13.5 | - |
| SC | 27.7 | 58.2 | 53.3 | 26.8 | - | 63.1 | 67.8 | 8.2 | - |
| PaLM 540B | | | | | | | | | |
| CoT | 56.9 | 73.9 | 79.0 | - | - | 79.9 | 77.8 | 63.0 | - |
| SC | 74.4 | 81.9 | 86.6 | 48.3 | - | 80.7 | 81.6 | 70.8 | - |
| text-davinci-002 | | | | | | | | | |
| Auto-CoT | 47.9 | - | 69.5 | 36.5 | 87.0 | 74.4 | 65.4 | 59.7 | - |
| CoT | 46.9 | 71.3 | 68.9 | 35.8 | 77.3 | 73.5 | 65.4 | 56.6 | 61.5 |
| SC | 58.2 | 76.9 | 78.2 | 41.8 | 87.2 | 72.9 | 70.7 | 57.6 | 67.9 |
| Active-Prompt (D) | 73.2 | 83.2 | 82.7 | 48.4 | 90.6 | 76.6 | 76.9 | 67.7 | 74.9 |
| code-davinci-002 | | | | | | | | | |
| Auto-CoT | 62.8 | - | - | - | - | - | - | - | - |
| CoT | 63.1 | 80.4 | 76.4 | 45.3 | 93.1 | 77.9 | 73.2 | 70.4 | 72.5 |
| SC | 78.0 | 87.8 | 86.8 | 52.0 | 93.7 | 81.5 | 79.8 | 73.4 | 79.1 |
| Random-CoT | 78.6 | 87.1 | 88.0 | 53.1 | 94.0 | 82.1 | 79.4 | 73.3 | 79.4 |
| Active-Prompt (D) | 82.2 | 88.4 | 88.7 | 55.1 | 94.5 | 83.9 | 80.6 | 74.1 | 80.9 |
| Active-Prompt (E) | 83.4 | 89.3 | 87.5 | 57.0 | 95.5 | 82.6 | 80.6 | 76.7 | 81.6 |

Table 2: The overall performance of Active-Prompt and the comparison against existing models on six downstream tasks. CoT and SC denote chain-of-thought (Wei et al., 2022b) and self-consistency (Wang et al., 2022c) methods. **Bold** denotes the best result. *a*: Cobbe et al. (2021), *b*: Lan et al. (2022), *c*: Pi et al. (2022), *d*: Amini et al. (2019), *e*: Hu et al. (2019), *f*: Xu et al. (2021), *g*: Chowdhery et al. (2022). Most statistics of Manual-CoT and SC are obtained directly from their latest version.

- 注意到人类提示工程有两个互补的组成部分：问题选择和提示模板工程。本文提出的方法很好地解决了第一个问题。还可以将此技术与人为优化的提示模板结合起来，以进一步提高性能。

Progressive-Hint Prompting Improves Reasoning in Large Language Models (渐进式提示提高大语言模型的推理能力)

Chuanyang Zheng Zhengying Liu Enze Xie Zhenguo Li Yu Li

Chinese University of Hong Kong, Huawei Noah's Ark Lab

汇报人：修娜娜

大型语言模型 (LLM) 在推理任务中的表现在很大程度上依赖于提示设计，而思维链 (CoT) 和self-consistency (SC) 是增强这种能力的关键方法。然而，这些方法并没有充分利用LLM生成的答案来指导后续的响应。

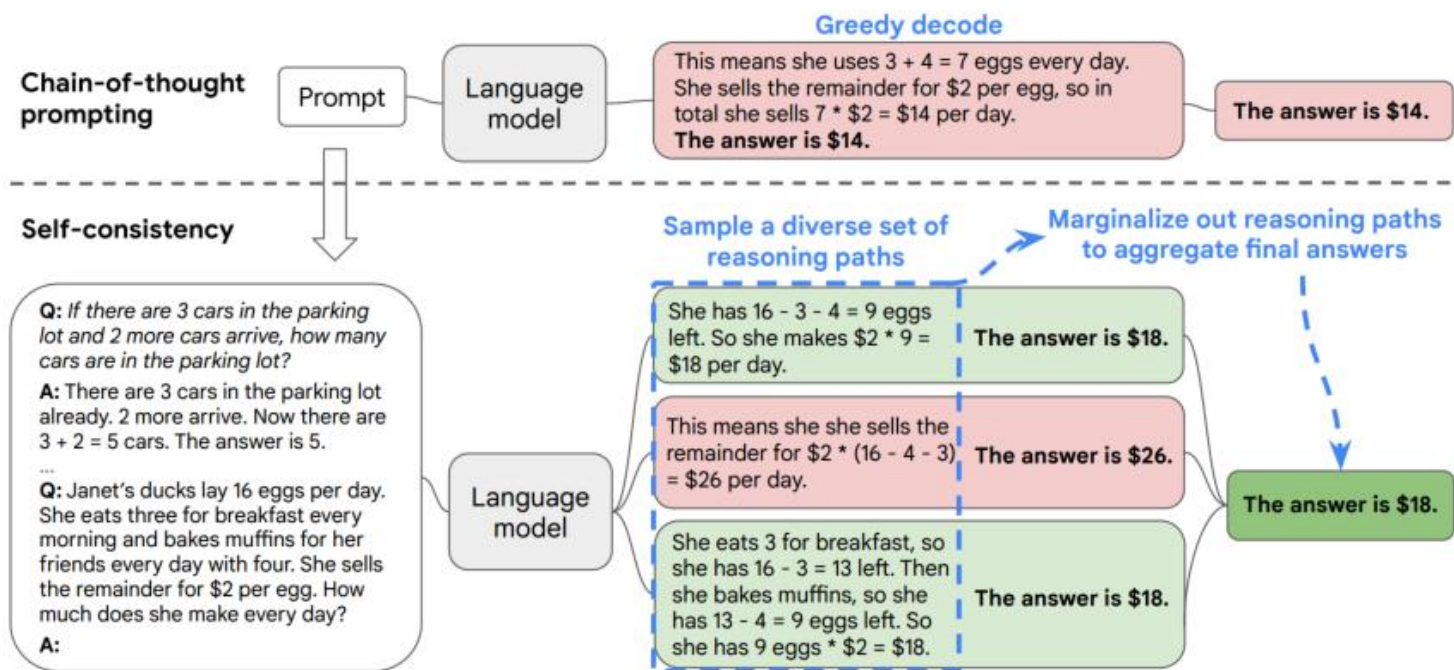


Figure 1: The self-consistency method contains three steps: (1) prompt a language model using chain-of-thought (CoT) prompting; (2) replace the “greedy decode” in CoT prompting by sampling from the language model’s decoder to generate a diverse set of reasoning paths; and (3) marginalize out the reasoning paths and aggregate by choosing the most consistent answer in the final answer set.

论文工作-背景

作者提出了一种新的提示方法，即渐进式提示(Progressive-Hint prompt, PHP)，通过使用之前生成的答案作为提示，逐步引导正确答案，实现用户与LLM之间的自动多次交互。PHP与CoT和self-consistency是正交的，这使得它很容易与最先进的技术相结合，从而进一步提高性能。

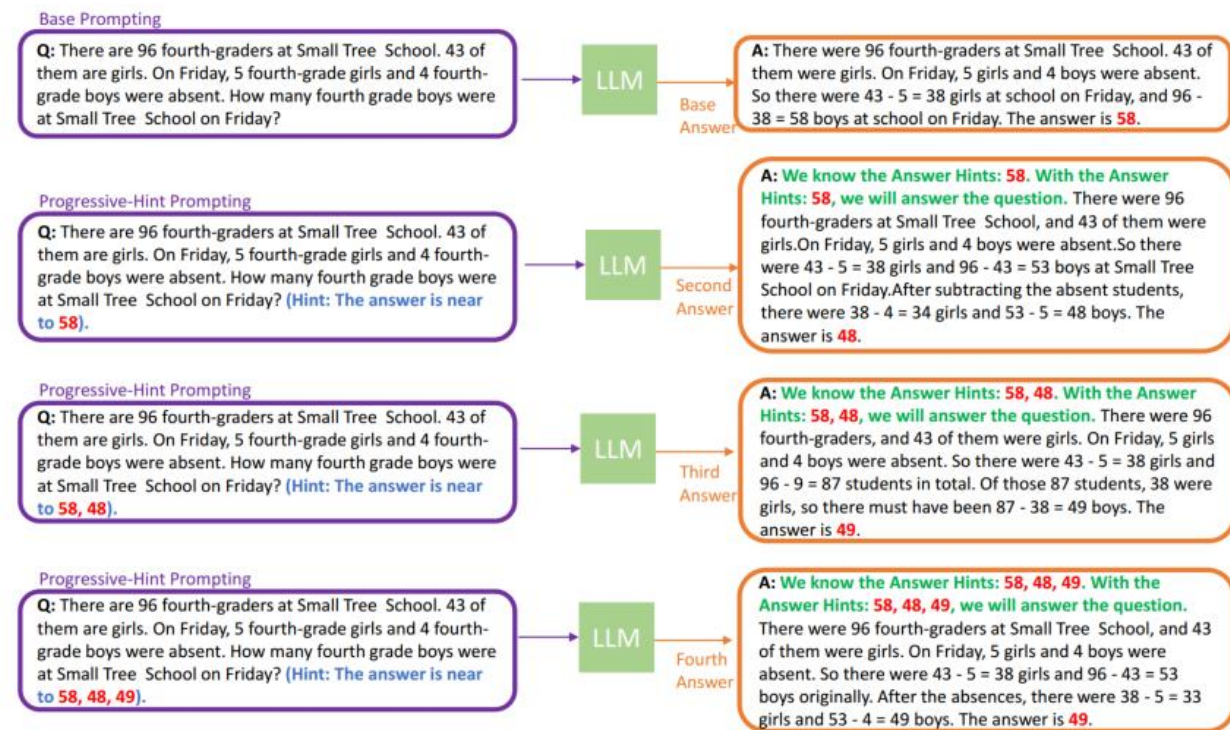
本文进行了广泛而全面的评估，以证明所建议方法的有效性。在六个基准测试上的实验结果表明，将CoT和self-consistency与PHP相结合可以显著提高准确性，同时保持高效率。

本文相信渐进式提示是与LLM自动顺序交互的第一步，并希望它能激发该领域未来的研究。

人类的一个突出方面是我们不仅能够思考一次，而且还能够反复检查我们的答案。作者提出这个过程可以在语言模型中通过顺序地使用先前的答案来模拟。

换句话说，一个模型可以生成一个答案，然后将其与下一轮思考的问题结合起来。如果当前的答案与之前的答案相同，我们就可以确信当前的答案是正确的。

Figure 1: Our proposed Progressive-Hint Prompting method combines the generated answers and questions for double-checking purposes, which is divided into two stages. In the first stage, we generate a **base answer** by passing to the LLM a concatenation of the current question and a **base prompt**, such as CoT or Complex CoT. In the second stage, we generate the **subsequent answers** via the corresponding **progressive-hint prompt**, such as Progressive-Hint Prompting CoT (PHP-CoT) or Progressive-Hint Prompting Complex CoT (PHP-Complex CoT), for the subsequent interaction. The interaction stops when two consecutive answers are the same. **Purple Box**: The input of LLM. **Orange Box**: The output of LLM.



operates as follows: (1) given a question, we ask the LLM to provide a Base Answer; (2) we combine the question and answer to re-ask the LLM and obtain the Subsequent Answer; (3) we repeat the operation in (2) until the answer converges and does not change over the last two answers. PHP follows a human-like thought process where previous answers are leveraged as hints to arrive at the correct answer after re-evaluating the question.

应该考虑各种hint的情况。当问LLM问题时不知道答案是什么，所以提示是未知的。在这个提示设计中，考虑了以下两种可能的情况：1) 提示与正确答案相同：确保当提示正确时，模型仍然可以得到正确答案；2) 提示与正确答案不一样：要确保模型能跳出错误答案。

| | |
|---|--|
| Table 1: Illustration of Progressive-Hint Prompting. Progressive-Hint Prompting CoT generation from the base prompt CoT. Given a base prompt (CoT as example), we create the corresponding Progressive-Hint prompt with two steps: 1) Add Answer Hint in the Question Phase; 2) Rehearse the Answer Hint in the Answer Phase. The answer hint in the Progressive-Hint prompt is the potential answer, and they could be replaced by other numbers. Blue Color : The difference between Base CoT and PHP-CoT. Red Color : The handcrafted Hint in the designed prompt. | |
| Base Prompting (e.g. CoT) | |
| Q: | There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? |
| A: | There are 15 trees originally. Then there were 21 trees after the Grove workers planted some more. So there must have been $21 - 15 = 6$ trees that were planted. The answer is 6. |
| Progressive-Hint Prompting 1: Hint is the correct answer | |
| Q: | There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? (Hint: The answer is near to 6). |
| A: | We know the Answer Hints: 6. With the Answer Hints: 6, we will answer the question. There are 15 trees originally. Then there were 21 trees after the Grove workers planted some more. So there must have been $21 - 15 = 6$ trees that were planted. The answer is 6. |
| Progressive-Hint Prompting 2: Hint is the incorrect answer | |
| Q: | There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? (Hint: The answer is near to 10, 8). |
| A: | We know the Answer Hints: 10, 8. With the Answer Hints: 10, 8, we will answer the question. There are 15 trees originally. Then there were 21 trees after the Grove workers planted some more. So there must have been $21 - 15 = 6$ trees that were planted. The answer is 6. |



Question: A shopkeeper in order to promote his new shop put a discount of 20% on all the items for one day. Now he must sell the items at original price the other day. By what percentage must he increase the price to original? Answer Choices: (A) 21% (B) 20% (C) 25% (D) 33% (E) 18%

A: Let's think step by step

The shopkeeper gave a discount of 20% on all the items.

Therefore, the new price of the items = 80% of the original price.

To get back to the original price, the shopkeeper must increase the price by $(100\% - 80\%) = 20\%$.

The answer is (B), 20%

Question: A shopkeeper in order to promote his new shop put a discount of 20% on all the items for one day. Now he must sell the items at original price the other day. By what percentage must he increase the price to original? Answer Choices: (A) 21% (B) 20% (C) 25% (D) 33% (E) 18% (Hint: The answer is near to (B)).

A: We know the Answer Hints: (B).

With the Answer Hints: (B), we will answer the question.

Let's think step by step

Let the original price of the item be x .

Then, the price after discount = $x - (20/100)x = 0.8x$

To get the original price, the shopkeeper must increase the price by $(100/80)\% = 25\%$.

The answer is (C), 25%

Question: A shopkeeper in order to promote his new shop put a discount of 20% on all the items for one day. Now he must sell the items at original price the other day. By what percentage must he increase the price to original? Answer Choices: (A) 21% (B) 20% (C) 25% (D) 33% (E) 18% (Hint: The answer is near to (B), (C)).

A: We know the Answer Hints: (B), (C).

With the Answer Hints: (B), (C), we will answer the question.

Let's think step by step

Let the original price of the item be x .

Then, the discounted price of the item = $x - (20/100)x = 0.8x$

To get the original price, the shopkeeper must increase the price by $(100/80)\% = 25\%$.

The answer is (C), 25%

- 当使用的提示和模型越强大，提出的PHP提高性能的表现越明显。

Table 2: PHP, when applied to different LLMs and prompting methods, can help to improve the performance. Meanwhile, PHP works better when the model and prompt are more powerful. The results are with greedy decoding.

| | Prompt | PHP | Dataset | | | | | Average | |
|----------------------------|------------------|-----|---------|------------|----------|--------|--------|---------|---------|
| | | | AddSub | MultiArith | SingleEQ | SVAMP | GSM8K | | AQuA |
| GPT3.5 text-davinci-002 | Standard [8] | ✗ | 79.4 | 34.0 | 80.7 | 64.8 | 15.1 | 25.5 | 49.91 |
| | | ✓ | 80.5 | 31.8 | 79.9 | 64.2 | 14.7 | 25.5 | 49.43 |
| | | | (+1.1) | (-2.2) | (-0.8) | (-0.6) | (-0.4) | (0.0) | (-0.48) |
| | CoT [8] | ✗ | 85.8 | 89.1 | 89.7 | 72.9 | 49.5 | 44.4 | 71.89 |
| | | ✓ | 86.8 | 89.0 | 90.1 | 72.3 | 51.1 | 45.6 | 72.48 |
| | | | (+1.0) | (-0.1) | (+0.4) | (-0.6) | (+1.6) | (+1.2) | (+0.59) |
| | Complex CoT [10] | ✗ | 82.5 | 89.8 | 87.7 | 70.4 | 57.6 | 37.4 | 70.89 |
| | | ✓ | 83.7 | 90.1 | 89.9 | 74.6 | 61.2 | 37.0 | 72.75 |
| | | | (+1.2) | (+0.3) | (+2.2) | (+4.2) | (+3.6) | (-0.4) | (+1.86) |
| GPT3.5 text-davinci-003 | Standard [8] | ✗ | 89.1 | 36.3 | 83.8 | 68.7 | 15.9 | 28.3 | 53.68 |
| | | ✓ | 89.1 | 36.0 | 83.6 | 68.7 | 16.0 | 28.3 | 53.61 |
| | | | (0.0) | (-0.3) | (-0.2) | (0.0) | (+0.1) | (0.0) | (-0.07) |
| | CoT [8] | ✗ | 90.6 | 93.6 | 92.7 | 81.0 | 56.1 | 44.0 | 76.33 |
| | | ✓ | 91.1 | 94.0 | 93.5 | 81.3 | 57.5 | 44.4 | 76.96 |
| | | | (+0.5) | (+0.4) | (+0.8) | (+0.3) | (+1.4) | (+0.4) | (+0.63) |
| | Complex CoT [10] | ✗ | 86.3 | 94.8 | 91.5 | 77.4 | 67.0 | 48.8 | 77.63 |
| | | ✓ | 88.1 | 95.0 | 94.0 | 80.0 | 71.6 | 50.0 | 79.78 |
| | | | (+1.8) | (+0.2) | (+2.5) | (+2.6) | (+4.6) | (+1.2) | (+2.15) |

- 模型越强大，prompt越弱，交互次数越少。当使用相同的模型时，交互次数通常随着提示的增强而增加。这是因为当提示变得更有效时，LLM获得了更好的推理能力，允许它们利用提示跳出不正确的答案，并最终导致获得最终答案所需的更多交互。

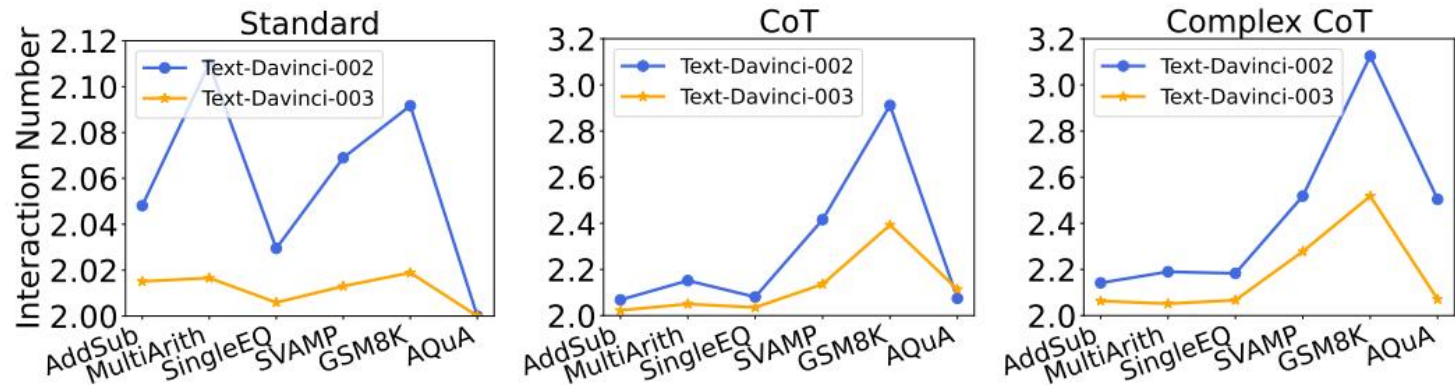


Figure 2: The Interaction Number refers to the frequency at which we need to consult the LLM until we receive conclusive responses. With an analysis of various models and prompts, it has been observed that: 1) A stronger model leads to a decreased interaction number; 2) An improved prompt results in an increased interaction number.

- 提示的质量会显著影响性能。为了增强PHP-Standard，将基本提示标准替换为Complex CoT或CoT，可以显著提高最终性能。
- 如果PHP不是根据相应的基本提示符设计的，性能可能会进一步提高。这相当于两个人合作解决一个问题。因此，在这种情况下，系统的性能可能会进一步提高。

Table 3: Performance with different Base Answers. Initially, the base prompt provides base answers to the model and PHP generates the subsequent answers. It is important to note that the quality of the employed hint significantly impacts the performance. Specifically, replacing the base prompt Standard with Complex CoT or CoT leads to a remarkable improvement in the final performance. Conversely, replacing the base prompt Complex CoT with Standard will reduce the final performance. The results are from text-davinci-003 with greedy decoding.

| PHP | Base Prompt | Dataset | | | | | | Average |
|-----------------|------------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|
| | | AddSub | MultiArith | SingleEQ | SVAMP | GSM8K | AQuA | |
| PHP-Standard | Standard [8] | 89.1 | 36.0 | 83.6 | 68.7 | 16.0 | 28.3 | 53.61 |
| | CoT [8] | 92.4 | 80.5 | 92.1 | 78.5 | 50.2 | 42.5 | 72.70 |
| | Complex CoT [10] | 90.6 | 80.6 | 92.9 | 77.2 | 60.3 | 45.6 | 74.53 |
| PHP-CoT | Standard [8] | 90.8 | 92.5 | 90.7 | 80.2 | 52.3 | 40.9 | 74.56 |
| | CoT [8] | 91.1 | 94.0 | 93.5 | 81.3 | 57.5 | 44.4 | 76.96 |
| | Complex CoT [10] | 90.6 | 96.8 | 93.7 | 81.2 | 62.6 | 50.0 | 79.14 |
| PHP-Complex CoT | Standard [8] | 88.3 | 80.1 | 93.3 | 80.4 | 65.5 | 35.4 | 73.83 |
| | CoT [8] | 88.8 | 95.6 | 94.8 | 81.4 | 70.6 | 45.6 | 79.46 |
| | Complex CoT [10] | 88.1 | 95.0 | 94.0 | 80.0 | 71.6 | 50.0 | 79.78 |

Table 4: A GSM8K example of text-davinci-003 response with greedy decoding, and the prompt is PHP-Complex CoT. Based on our analysis, it appears that a hint that is too far from the correct answer has the potential to mislead LLMs and lead to an incorrect final answer. Conversely, a more effective hint can result in a more precise outcome.

Question: 200 pounds of carrots will be distributed to 40 restaurants in a certain city. Each restaurant is to receive 2 pounds of carrots. How many pounds of carrots will not be used?

Hint: 0. We know the Answer Hints: 0. With the Answer Hints: 0, we will answer the question. Let's think step by step We know that 200 pounds of carrots are to be distributed to 40 restaurants. Each restaurant is to receive 2 pounds of carrots. Therefore, $40 \text{ restaurants} \times 2 \text{ pounds of carrots} = 80 \text{ pounds of carrots}$. The total amount of carrots used is $200 \text{ pounds} - 80 \text{ pounds} = 120 \text{ pounds}$. The amount of carrots not used is $200 \text{ pounds} - 120 \text{ pounds} = 80 \text{ pounds}$. The answer is 0

Hint: 100. We know the Answer Hints: 100. With the Answer Hints: 100, we will answer the question. Let's think step by step We know that 200 pounds of carrots are to be distributed to 40 restaurants. Each restaurant will receive 2 pounds of carrots, so $40 \text{ restaurants} \times 2 \text{ pounds} = 80 \text{ pounds of carrots}$. This means that $200 \text{ pounds of carrots} - 80 \text{ pounds of carrots} = 120 \text{ pounds of carrots}$ will not be used. The answer is 120

Performance with chat model:

本文提供两个例子来说明这一点:

a) 在没有给出提示的情况下, GPT-3.5-Turbo不能回答出答案: We cannot answer this question as the answer hint is missing. Please provide the answer hint to proceed. 相比之下, text-davinci-003在回答问题之前会自动生成并填写缺失的答案提示;

b) 当提供超过10个提示时, GPT-3.5-Turbo可能会回应 “We cannot determine the correct answer as multiple answer hints are given. Please provide only one answer hint for the question.”。请只提供一个问题的答案提示。在text-davinci-003中没有观察到这种行为。因此, OpenAI可能会采用不同的RLHF训练数据或实施替代技术, 以赋予GPT-3.5-Turbo更大的响应灵活性

分析GPT-4。在部署GPT-4模型后，本文的方法能够在SVAMP、GSM8K和AQuA基准测试中实现新的SOTA性能。PHP方法持续提高了GPT-4的性能，平均提高了0.83%。此外，与GPT-3.5-Turbo模型相比，GPT-4所需的相互作用次数减少，这与“模型更强大时相互作用次数减少”的发现一致。

Table 8: Performance of Complex CoT with GPT-3.5-Turbo and GPT-4, employing greedy decoding. Number: The average interaction number with LLM.

| | PHP | Dataset | | | | | | Average |
|------------------|--------|---------|------------|----------|--------|--------|--------|---------|
| | | AddSub | MultiArith | SingleEQ | SVAMP | GSM8K | AQuA | |
| GPT-3.5 Turbo | ✗ | 85.5 | 97.5 | 92.5 | 81.0 | 82.8 | 57.4 | 82.78 |
| | ✓ | 85.3 | 98.0 | 92.9 | 83.1 | 85.1 | 60.6 | 84.16 |
| | | (-0.2) | (+0.5) | (+0.4) | (+2.1) | (+2.3) | (+3.2) | (+1.38) |
| | Number | 2.1037 | 2.0133 | 2.0610 | 2.3570 | 2.3426 | 2.3228 | 2.2000 |
| GPT-4 | ✗ | 89.3 | 97.8 | 93.1 | 90.5 | 94.9 | 77.5 | 90.51 |
| | ✓ | 89.6 | 98.1 | 93.1 | 91.9 | 95.5 | 79.9 | 91.34 |
| | | (+0.3) | (+0.3) | (0.0) | (+1.4) | (+0.6) | (+2.4) | (+0.83) |
| | Number | 2.0126 | 2.0033 | 2.0019 | 2.0700 | 2.0507 | 2.2913 | 2.0716 |

- 为了更好地完善渐进式提示方法，今后的研究可以着重改进提问环节的手工提示设计和答案环节的提示句设计。
- 此外，可以在答案旁边识别和提取可以帮助LLM重新考虑问题的新提示。

THANKS