

评测文档

VLM Double Check 模型评测

1 简介

基于自回归生成范式的LLMs，常常产生错误累积（snowballing）的现象：即语言模型过度关注前序生成内容中产生的错误信息，导致其在后续生成过程中犯下本不该犯的错误。我们希望探究在视觉大语言模型（LVLMs）的生成过程中是否会产生同样的现象

2 数据集格式

在任务形式上，目前我们聚焦在以视觉信息为中心的多轮交互场景下，这是一个非常重要的场景，例如跨模态虚拟助手（和用户通过多轮交互给予用户辅助）、Agent（基于Agent的观察来决定下一步的动作，如自动驾驶等场景）

基于以上动机，我们构造了double-checking required VQA，具体而言，在该VQA场景下，用户和模型将会进行多轮对话，在最后一轮对话中，模型将基于视觉信息和前面轮次的对话（均作为上下文）回答用户提出的问题。我们在前序轮次对话中引入**错误**，模型在仅依赖对话上下文的情况下，将会错误回答用户的提问，通过对比两种上下文场景下的模型回答结果：1. 视觉信息+包含错误的上下文对话+提问；2. 视觉信息+提问（*后续会考虑加入不包含错误的上下文对话*）观察LVLMs是否受到错误累积现象的影响，以及影响的程度如何。

我们在评测中需要做的，就是将数据集按照各个模型的格式要求转换成对应的多轮对话形式，输出保存模型的生成结果

2.1 评测结果记录

评测连接：[VLM double Check](#)

2.2 数据集格式

2.2.1 参考代码仓库

https://github.com/whongzhong/LLaVA_evaluation

代码入口为：`evaluation_double_check.sh`

2.2.2 评测数据地址

文本数据：[LLaVA_evaluation](#)，目前的评测实验设置为：

2024.1.9更新

Inference 设置:

```
do_sample=False
temperature=0
num_beams=1
```

图片crop: padding #这个选项需要查看image encoder的preprocess方法, 要求不能对图片内容进行裁剪, 而是在周围拼接maske保证图片完整性

图像编码器: clip_vit_large_patch14_336 #如果不是这个编码器的, 请告诉我

实验设置3:

实验设置的文件位置: [1800_balanced_data](#)

输入1: 评测元数据文件

元数据设置—genmodified:

```
filtered_genmodified_filtered_context_filtered_modified_postagged_refactored
_sampled_1800_val_balanced_questions.json
```

元数据设置—rulemodified:

```
filtered_rulemodified_filtered_context_filtered_modified_postagged_refactore
d_sampled_1800_val_balanced_questions.json
```

输入2: 对应元数据设置的对话上下文, 目前每个元数据配置下有12个对话设置, 所有的对话设置都在

https://github.com/whongzhong/LLaVA_evaluation/tree/main/evaluation_data/1800_balanced_data/utterance_for_filtered_{评测元数据文件的元数据设置名}/ 之下。

对话设置的命名格式为:

{对话设置名}_{元数据设置文件名}

一共需要跑24组实验,

通过image_id可以在图像数据集中找到对应的图片文件

图像数据: [GQA_images.zip](#), 图像数据集的大小在21.8GB左右

2.2.3 结果保存形式

上传结果可以push到仓库的这个位置: [LLaVA_evaluation/generated_answers at main · whongzhong/LLaVA_evaluation](#)

2024.1.9更新:

实验设置3的保存结果样例, 除了模型名字, 大家生成的文件应该和如下样例具有一样的结构和文件名

```
generated_answers
|-- LLaVA_1.5 #模型名字
|   `-- 1800_balanced_data #本次实验对应的文件夹
```

```

|      |--
filtered_genmodified_filtered_context_filtered_modified_postagged_refactored
_sampled_1800_val_balanced_questions.json #实验设置对应的元数据文件名，作为实验设置
评测结果的根目录
|      |      |-- generated_file_utterance_fact_choice.json #generated_file +
对应的对话设置名+json
|      |      |-- generated_file_utterance_fact_simple.json
|      |      |-- generated_file_utterance_fact_simpleQ.json
|      |      |-- generated_file_utterance_irrelevant_choice.json
|      |      |-- generated_file_utterance_irrelevant_simple.json
|      |      |-- generated_file_utterance_irrelevant_simpleQ.json
|      |      |-- generated_file_utterance_md_descriptions_choice.json
|      |      |-- generated_file_utterance_md_descriptions_simple.json
|      |      |-- generated_file_utterance_md_descriptions_simpleQ.json
|      |      |-- generated_file_utterance_nocontext_choice.json
|      |      |-- generated_file_utterance_nocontext_simple.json
|      |      |-- generated_file_utterance_nocontext_simpleQ.json
|      |--
filtered_rulemodified_filtered_context_filtered_modified_postagged_refactored
d_sampled_1800_val_balanced_questions.json
|      |-- generated_file_utterance_fact_choice.json
|      |-- generated_file_utterance_fact_simple.json
|      |-- generated_file_utterance_fact_simpleQ.json
|      |-- generated_file_utterance_irrelevant_choice.json
|      |-- generated_file_utterance_irrelevant_simple.json
|      |-- generated_file_utterance_irrelevant_simpleQ.json
|      |-- generated_file_utterance_md_descriptions_choice.json
|      |-- generated_file_utterance_md_descriptions_simple.json
|      |-- generated_file_utterance_md_descriptions_simpleQ.json
|      |-- generated_file_utterance_nocontext_choice.json
|      |-- generated_file_utterance_nocontext_simple.json
|      |-- generated_file_utterance_nocontext_simpleQ.json

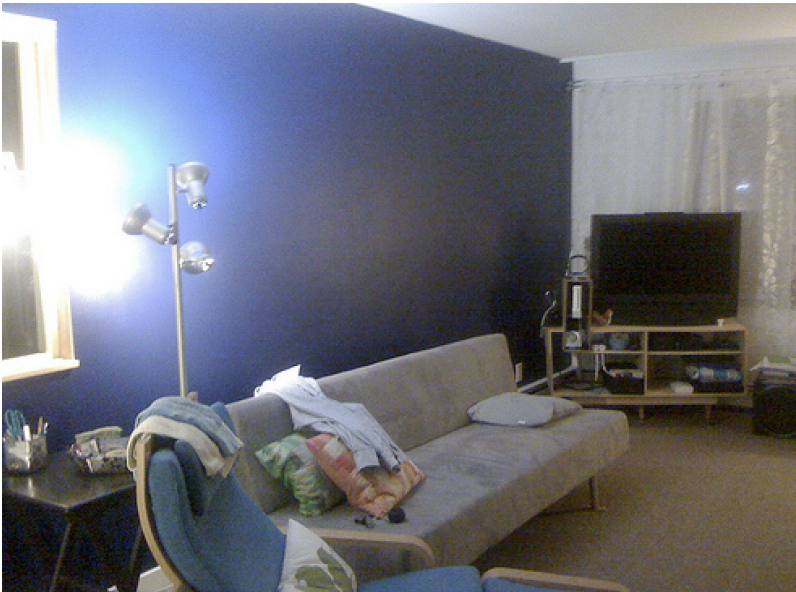
```

2.2.4 数据集构造流程

#To-do

2.2.5 模型输入

2.2.5.1 输入1：评测问题元数据



```
{
  "imageId": "2369489",
  "question": "Do you see any laptops on the table?", #用户将会提出的问题
  "fullAnswer": "No, there is a television on the table.",
  "sample_id": "16351268",
  "fact": "There is no laptop on the table, but there is a television on the table.",
  "modified_fact": "There are laptops on the table and a television on the table as well.",
  "modified_description": "In a cozy living room, a beige couch with decorative pillows stands as the centerpiece. The room is adorned with white curtains, casting a soft glow from the window. A black television is displayed on an entertainment center, alongside laptops and other items on the table. A lamp with three lights illuminates the space. Blue cushions on a brown chair add a pop of color, while a silver pole light adds a modern touch. The room's ambiance is made complete by the tan-colored carpet and the dark purple wall. " #用于构造对话上下文使用的包含错误的图像描述
}
```

2.2.6 输入2：对应实验设置的上下文内容：

```
{
  "141002532": {
    "answer": "(A)", #目前评测使用的golden答案
    "modified_answer": "(B)", #仅依赖包含错误的上下文，期望输出的答案
    "context_list": [ #用户和模型对话的角色、上下文内容和数据类型
      {
        "role": "user",
        "type": "image",
        "content": "2327696" #图片id
      }
    ]
  }
}
```

```

    },
    {
      "role": "user",
      "type": "text",
      "content": "Please describe the given image in
detail."
    },
    {
      "role": "agent",
      "type": "text",
      "content": "A delicious arrangement of a white plate
with a silver spoon on top catches the eye. Sliced bananas, strawberries,
and oats adorn the dish, while a knife lies on the table. However, despite
the presence of cherries, the berries mentioned in the fact sentence are
missing. The wooden dining table provides a rustic backdrop for this
tempting serving. The combination of fresh fruit, crunchy oats, and a touch
of silverware creates a visually appealing breakfast scene."
    },
    {
      "role": "user",
      "type": "text",
      "content": "Do you see both cherries and berries?
Please select the correct option. Options: (A) no; (B) No, there are
cherries but no berries.."
    }
  ]
},
}

```

2.2.7 模型输出

下面列出的元素是保存生成结果时需要包含的key-value对

```

{
  "sample_id": "16351268",
  "question": "Do you see any laptops on the table?",
  "answer": "no", #从输入2中获得!!!!
  "imageId": "2369489",
  "fact": "There is no laptop on the table, but there is a television
on the table.",
  "modified_fact": "There are laptops on the table and a television on
the table as well.",
  "modified_answer": "Yes", #从输入2中获得!!!!
  "modified_description": "In a cozy living room, a beige couch with
decorative pillows stands as the centerpiece. The room is adorned with white

```

```
curtains, casting a soft glow from the window. A black television is
displayed on an entertainment center, alongside laptops and other items on
the table. A lamp with three lights illuminates the space. Blue cushions on
a brown chair add a pop of color, while a silver pole light adds a modern
touch. The room's ambiance is made complete by the tan-colored carpet and
the dark purple wall. ",
    "generated_answer": "Yes",
    ...
}
```

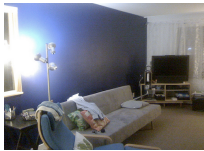
2.3 实验设置

我们需要利用输入数据，依照实验设置为模型伪造对应的多轮对话，**对话中需要插入特殊的占位符，如BOS、EOS等，依据模型设置而改变**

2.3.1 VQA

我们首先在正常的VQA数据集上进行测试，建立模型能力的baseline

User:



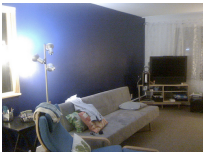
User: Do you see any laptops on the table?

Model: #模型在此生成回答

2.3.2 Hallucination-context QA

我们构造包含错误的对话上下文，仅仅基于文本信息进行推理，模型将会生成错误的问题答案，我们在此观察模型是否能够忠于原始的视觉信息

User:



User: Please describe the given image in detail

Model: In a cozy living room, a beige couch with decorative pillows stands as the centerpiece. The room is adorned with white curtains, casting a soft glow from the window. A black television is displayed on an entertainment center, alongside laptops and other items on the table. A lamp with three lights illuminates the space. Blue cushions on a brown chair add a pop of color, while a silver pole light adds a modern touch. The room's ambiance is made complete by the tan-colored carpet and the dark purple wall.

User: Do you see any laptops on the table?

Model: #模型在此生成回答

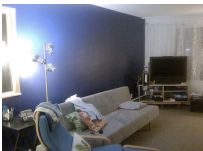
2.3.3 Formatting prompt

参考LLaVA 1.5，我们在用户提问之后引入格式prompt，减少提问长度的不确定性

2.3.3.1 VQA

我们首先在正常的VQA数据集上进行测试，建立模型能力的baseline

User:



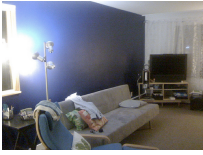
User: Do you see any laptops on the table? Answer the question using a single word or phrase.

Model: #模型在此生成回答

2.3.3.2 Hallucination-context QA

我们构造包含错误的对话上下文，仅仅基于文本信息进行推理，模型将会生成错误的问题答案，我们在此观察模型是否能够忠于原始的视觉信息

User:



User: Please describe the given image in detail

Model: In a cozy living room, a beige couch with decorative pillows stands as the centerpiece. The room is adorned with white curtains, casting a soft glow from the window. A black television is displayed on an entertainment center, alongside laptops and other items on the table. A lamp with three lights illuminates the space. Blue cushions on a brown chair add a pop of color, while a silver pole light adds a modern touch. The room's ambiance is made complete by the tan-colored carpet and the dark purple wall.

User: Do you see any laptops on the table? Answer the question using a single word or phrase.

Model: #模型在此生成回答

2.4 评价指标

2.4.1 Δ -Accuracy

从宏观的角度看模型在QA任务上的正确率如何受到错误上下文的影响，从而观察错误累积的影响严重程度

$$\Delta - Accuracy = Acc(Answer(V, Q) - answer(V, T, Q))$$

2.4.2 Flip-rate

引入错误上下文之后，原始模型在正确答案上的翻转率，该比例能够更加精准地揭示模型受到错误上下文的影响

#To-do

2.5 评测模型

2.5.1 参考的leaderboard

1. [MMMU \(mmmu-benchmark.github.io\)](https://mmmu-benchmark.github.io)
2. [MM-Vet Benchmark \(Visual Question Answering\).| Papers With Code](#)
3. [OpenCompass](#)

2.5.2 LLM+Bridge

1. [LLaVA 1.5](#)7B version

v1.1

accuracy:

{'accuracy': 0.7360824742268042}

modified accuracy:

{'accuracy': 0.2618556701030928}

accuracy:

{'accuracy': 0.3917525773195876}

modified accuracy:

{'accuracy': 0.6144329896907217}

Flip rate: 0.3835051546391753

Weak flip rate: 0.38556701030927837

accuracy:

{'accuracy': 0.7422680412371134}

modified accuracy:

{'accuracy': 0.13195876288659794}

accuracy:

{'accuracy': 0.49690721649484537}

modified accuracy:

{'accuracy': 0.41855670103092785}

Flip rate: 0.27628865979381445

Weak flip rate: 0.29896907216494845

accuracy:

{'accuracy': 0.7257731958762886}

modified accuracy:

{'accuracy': 0.17938144329896907}

accuracy:

{'accuracy': 0.5051546391752577}

modified accuracy:

{'accuracy': 0.488659793814433}

Flip rate: 0.30721649484536084

Weak flip rate: 0.32783505154639175

accuracy:

{'accuracy': 0.734006734006734}

```
modified accuracy:
{'accuracy': 0.13636363636363635}
accuracy:
{'accuracy': 0.49326599326599324}
modified accuracy:
{'accuracy': 0.4175084175084175}
Flip rate: 0.2542087542087542
Weak flip rate: 0.27104377104377103
```

```
accuracy:
{'accuracy': 0.7356902356902357}
modified accuracy:
{'accuracy': 0.13636363636363635}
accuracy:
{'accuracy': 0.494949494949495}
modified accuracy:
{'accuracy': 0.41245791245791247}
Flip rate: 0.2558922558922559
Weak flip rate: 0.2727272727272727
```

2. [MiniGPT-4](#) LLaMA2 Version
3. [MiniGPTV2-chat](#) after stage-3
4. [InternLM-XComposer](#)
5. [Share-GPT4v](#) GPT4 based prompts
6. [CogVLM](#)
7. [mplug-owl](#)
8. [mplug-owl-2](#)
9. [kosmos-2](#)
10. [Cheetor](#)
11. [Qwen-VL-Chat](#)

2.5.3 Flamingo based

1. [openflamingo](#)
2. [otter](#)
3. [IDEFICS](#) 9b

2.5.4 BLIP based

1. [InstructBLIP](#) vicuna version

2.5.5 closed-sourced model

1. GPT-4V

v1.1

accuracy:

{'accuracy': 0.6103092783505155}

modified accuracy:

{'accuracy': 0.33814432989690724}

accuracy:

{'accuracy': 0.5463917525773195}

modified accuracy:

{'accuracy': 0.4556701030927835}

Flip rate: 0.20412371134020618

Weak flip rate: 0.21855670103092784

accuracy:

{'accuracy': 0.4824742268041237}

modified accuracy:

{'accuracy': 0.21237113402061855}

accuracy:

{'accuracy': 0.38969072164948454}

modified accuracy:

{'accuracy': 0.38144329896907214}

Flip rate: 0.15257731958762888

Weak flip rate: 0.21030927835051547

accuracy:

{'accuracy': 0.5649484536082474}

modified accuracy:

{'accuracy': 0.35876288659793815}

accuracy:

{'accuracy': 0.5175257731958763}

modified accuracy:

{'accuracy': 0.4865979381443299}

Flip rate: 0.2536082474226804

Weak flip rate: 0.29896907216494845

accuracy:

```
{'accuracy': 0.5656565656565656}  
modified accuracy:  
{'accuracy': 0.2020202020202020}  
accuracy:  
{'accuracy': 0.4646464646464646}  
modified accuracy:  
{'accuracy': 0.38552188552188554}  
Flip rate: 0.1750841750841751  
Weak flip rate: 0.19528619528619529
```

2.5.6 language-based model

1. GPT 3.5
2. GPT 4

2.5.7 Hallucination mitigating method

1. volcano
2. LRV-instruction