大模型评测教程

随着人工智能技术的快速发展，大规模预训练自然语言模型成为了研究热点和关注焦点。OpenAI于2018年提出了第一代GPT模型，开辟了自然语言模型生成式预训练的路线。沿着这条路线，随后又陆续发布了GPT-2和GPT-3模型。与此同时，谷歌也探索了不同的大规模预训练模型方案，例如如T5，Flan等。OpenAI在2022年11月发布ChatGPT，展示了强大的问答能力，逻辑推理能力和内容创作能力，将模型提升到了实用水平，改变人们对大模型能力的认知。在2023年4月，OpenAI发布了新升级的GPT-4模型，通过引入多模态能力，进一步拓展了大语言模型的能力边界，朝着通用人工智能更进一步。ChatGPT和GPT-4推出之后，微软凭借强大的产品化能力迅速将其集成进搜索引擎和Office办公套件中，形成了New Bing和 Office Copilot等产品。谷歌也迅速上线了基于自家大语言模型PaLM和PaLM-2的Bard，与OpenAI和微软展开正面竞争。国内的多家企业和研究机构也在开展大模型的技术研发，百度，阿里，华为，商汤，讯飞等都发布了各自的国产语言大模型，清华，复旦等高校也相继发布了GLM, MOSS等模型。

为了准确和公正地评估大模型的能力，国内外机构在大模型评测上开展了大量的尝试和探索。斯坦福大学提出了较为系统的评测框架HELM，从准确性，安全性，鲁棒性和公平性等维度开展模型评测。纽约大学联合谷歌和Meta提出了SuperGLUE评测集，从推理能力，常识理解，问答能力等方面入手，构建了包括8个子任务的大语言模型评测数据集。加州大学伯克利分校提出了MMLU测试集，构建了涵盖高中和大学的多项考试，来评估模型的知识能力和推理能力。谷歌也提出了包含数理科学，编程代码，阅读理解，逻辑推理等子任务的评测集Big-Bench，涵盖200多个子任务，对模型能力进行系统化的评估。在中文评测方面，国内的学术机构也提出了如CLUE,CUGE等评测数据集，从文本分类，阅读理解，逻辑推理等方面评测语言模型的中文能力。

随着大模型的蓬勃发展，如何全面系统地评估大模型的各项能力成为了亟待解决的问题。由于大语言模型和多模态模型的能力强大，应用场景广泛，目前学术界和工业界的评测方案往往只关注模型的部分能力维度，缺少系统化的能力维度框架与评测方案。OpenCompass提供设计一套全面、高效、可拓展的大模型评测方案，对模型能力、性能、安全性等进行全方位的评估。OpenCompass提供分布式自动化的评测系统，支持对(语言/多模态)大模型开展全面系统的能力评估。

OpenCompass介绍

评测对象

本算法库的主要评测对象为语言大模型与多模态大模型。我们以语言大模型为例介绍评测的具体模型类型。

- 基座模型：一般是经过海量的文本数据以自监督学习的方式进行训练获得的模型（如OpenAI的GPT-3，Meta的LLaMA），往往具有强大的文字续写能力。
- 对话模型：一般是在的基座模型的基础上，经过指令微调或人类偏好对齐获得的模型（如OpenAI的ChatGPT、上海人工智能实验室的书生·浦语），能理解人类指令，具有较强的对话能力。

工具架构

- 模型层：大模型评测所涉及的主要模型种类，OpenCompass以基座模型和对话模型作为重点评测对象。
- 能力层：OpenCompass从本方案从通用能力和特色能力两个方面来进行评测维度设计。在模型通用能力方面，从语言、知识、理解、推理、安全等多个能力维度进行评测。在特色能力方面，从长文本、代码、工具、知识增强等维度进行评测。
- 方法层：OpenCompass采用客观评测与主观评测两种评测方式。客观评测能便捷地评估模型在具有确定答案（如选择，填空，封闭式问答等）的任务上的能力，主观评测能评估用户对模型回复的真实满意度，OpenCompass采用基于模型辅助的主观评测和基于人类反馈的主观评测两种方式。
- 工具层：OpenCompass提供丰富的功能支持自动化地开展大语言模型的高效评测。包括分布式评测技术，提示词工程，对接评测数据库，评测榜单发布，评测报告生成等诸多功能。

能力维度

设计思路

为准确、全面、系统化地评估大语言模型的能力，OpenCompass从通用人工智能的角度出发，结合学术界的前沿进展和工业界的最佳实践，提出一套面向实际应用的模型能力评价体系。OpenCompass能力维度体系涵盖通用能力和特色能力两大部分。

通用能力涵盖学科综合能力、知识能力、语言能力、理解能力、推理能力、安全能力，共计六大维度构造立体全面的模型能力评价体系。

特色能力

评测方法

OpenCompass采取客观评测与主观评测相结合的方法。针对具有确定性答案的能力维度和场景，通过构造丰富完善的评测集，对模型能力进行综合评价。针对体现模型能力的开放式或半开放式的问题、模型安全问题等，采用主客观相结合的评测方式。

客观评测

针对具有标准答案的客观问题，我们可以我们可以通过使用定量指标比较模型的输出与标准答案的差异，并根据结果衡量模型的性能。同时，由于大语言模型输出自由度较高，在评测阶段，我们需要对其输入和输出作一定的规范和设计，尽可能减少噪声输出在评测阶段的影响，才能对模型的能力有更加完整和客观的评价。

为了更好地激发出模型在题目测试领域的能力，并引导模型按照一定的模板输出答案，OpenCompass采用提示词工程 （prompt engineering）和语境学习（in-context learning）进行客观评测。

在客观评测的具体实践中，我们通常采用下列两种方式进行模型输出结果的评测：

- 判别式评测：该评测方式基于将问题与候选答案组合在一起，计算模型在所有组合上的困惑度（perplexity），并选择困惑度最小的答案作为模型的最终输出。例如，若模型在 `问题？ 答案1` 上的困惑度为 0.1，在 `问题？ 答案2` 上的困惑度为 0.2，最终我们会选择 `答案1` 作为模型的输出。

- 生成式评测：该评测方式主要用于生成类任务，如语言翻译、程序生成、逻辑分析题等。具体实践时，使用问题作为模型的原始输入，并留白答案区域待模型进行后续补全。我们通常还需要对其输出进行后处理，以保证输出满足数据集的要求。

主观评测

语言表达生动精彩，变化丰富，大量的场景和能力无法凭借客观指标进行评测。针对如模型安全和模型语言能力的评测，以人的主观感受为主的评测更能体现模型的真实能力，并更符合大模型的实际使用场景。

OpenCompass采取的主观评测方案是指借助受试者的主观判断对具有对话能力的大语言模型进行能力评测。在具体实践中，我们提前基于模型的能力维度构建主观测试问题集合，并将不同模型对于同一问题的不同回复展现给受试者，收集受试者基于主观感受的评分。由于主观测试成本高昂，本方案同时也采用使用性能优异的大语言模拟人类进行主观打分。在实际评测中，本文将采用真实人类专家的主观评测与基于模型打分的主观评测相结合的方式开展模型能力评估。

在具体开展主观评测时，OpenComapss采用单模型回复满意度统计和多模型满意度比较两种方式开展具体的评测工作。

快速开始

概览

在 OpenCompass 中评估一个模型通常包括以下几个阶段：配置 -> 推理 -> 评估 -> 可视化。

配置：这是整个工作流的起点。您需要配置整个评估过程，选择要评估的模型和数据集。此外，还可以选择评估策略、计算后端等，并定义显示结果的方式。

推理与评估：在这个阶段，OpenCompass 将会开始对模型和数据集进行并行推理和评估。推理阶段主要是让模型从数据集产生输出，而评估阶段则是衡量这些输出与标准答案的匹配程度。这两个过程会被拆分为多个同时运行的“任务”以提高效率，但请注意，如果计算资源有限，这种策略可能会使评测变得更慢。

可视化：评估完成后，OpenCompass 将结果整理成易读的表格，并将其保存为 CSV 和 TXT 文件。你也可以激活飞书状态上报功能，此后可以在飞书客户端中及时获得评测状态报告。

接下来，我们将展示 OpenCompass 的基础用法，展示书生浦语在 C-Eval 基准任务上的评估。它们的配置文件可以在 configs/eval_demo.py 中找到。

安装

面向GPU的环境安装

```
conda create --name opencompass --clone=/root/share/conda_envs/internlm-base
source activate opencompass
git clone https://github.com/open-compass/opencompass
cd opencompass
pip install -e .
```

有部分第三方功能,如代码能力基准测试 Humaneval 以及 Llama格式的模型评测,可能需要额外步骤才能正常运行，如需评测，详细步骤请参考安装指南。

数据准备

```
# 解压评测数据集到 data/ 处
cp /share/temp/datasets/OpenCompassData-core-20231110.zip /root/opencompass/
unzip OpenCompassData-core-20231110.zip

# 将会在opencompass下看到data文件夹
```

查看支持的数据集和模型

```
# 列出所有跟 internlm 及 ceval 相关的配置
python tools/list_configs.py internlm ceval
```

将会看到

```
+--------------------------+--------------------------------------------------
------+
| Model                    | Config Path
     |
|--------------------------+--------------------------------------------------
------|
| hf_internlm_20b          | configs/models/hf_internlm/hf_internlm_20b.py
     |
| hf_internlm_7b           | configs/models/hf_internlm/hf_internlm_7b.py
      |
| hf_internlm_chat_20b     |
configs/models/hf_internlm/hf_internlm_chat_20b.py     |
| hf_internlm_chat_7b      | configs/models/hf_internlm/hf_internlm_chat_7b.py
     |
| hf_internlm_chat_7b_8k   |
configs/models/hf_internlm/hf_internlm_chat_7b_8k.py   |
| hf_internlm_chat_7b_v1_1 |
configs/models/hf_internlm/hf_internlm_chat_7b_v1_1.py |
| internlm_7b              | configs/models/internlm/internlm_7b.py
      |
| ms_internlm_chat_7b_8k   |
configs/models/ms_internlm/ms_internlm_chat_7b_8k.py   |
+--------------------------+--------------------------------------------------
------+
+----------------------------+------------------------------------------------
------+
| Dataset                    | Config Path
     |
|----------------------------+------------------------------------------------
------|
| ceval_clean_ppl            | configs/datasets/ceval/ceval_clean_ppl.py
     |
| ceval_gen                  | configs/datasets/ceval/ceval_gen.py
     |
```

```
| ceval_gen_2daf24           | configs/datasets/ceval/ceval_gen_2daf24.py
      |
| ceval_gen_5f30c7           | configs/datasets/ceval/ceval_gen_5f30c7.py
      |
| ceval_ppl                  | configs/datasets/ceval/ceval_ppl.py
     |
| ceval_ppl_578f8d           | configs/datasets/ceval/ceval_ppl_578f8d.py
      |
| ceval_ppl_93e5ce           | configs/datasets/ceval/ceval_ppl_93e5ce.py
      |
| ceval_zero_shot_gen_bd40ef |
configs/datasets/ceval/ceval_zero_shot_gen_bd40ef.py |
+----------------------------+------------------------------------------------
------+
```

启动评测

确保按照上述步骤正确安装 OpenCompass 并准备好数据集后，可以通过以下命令评测 InternLM-Chat-7B 模型在 C-Eval 数据集上的性能。由于 OpenCompass 默认并行启动评估过程，我们可以在第一次运行时以 `--debug` 模式启动评估，并检查是否存在问题。在 `--debug` 模式下，任务将按顺序执行，并实时打印输出。

```
python run.py --datasets ceval_gen --hf-path /share/temp/model_repos/internlm-
chat-7b/ --tokenizer-path /share/temp/model_repos/internlm-chat-7b/ --
tokenizer-kwargs padding_side='left' truncation='left' trust_remote_code=True
--model-kwargs trust_remote_code=True device_map='auto' --max-seq-len 2048 --
max-out-len 16 --batch-size 4 --num-gpus 1 --debug
```

命令解析

```
--datasets ceval_gen \
--hf-path /share/temp/model_repos/internlm-chat-7b/ \  # HuggingFace 模型路径
--tokenizer-path /share/temp/model_repos/internlm-chat-7b/ \  # HuggingFace
tokenizer 路径（如果与模型路径相同，可以省略）
--tokenizer-kwargs padding_side='left' truncation='left'
trust_remote_code=True \  # 构建 tokenizer 的参数
--model-kwargs device_map='auto' trust_remote_code=True \  # 构建模型的参数
--max-seq-len 2048 \  # 模型可以接受的最大序列长度
--max-out-len 16 \  # 生成的最大 token 数
--batch-size 2  \  # 批量大小
--num-gpus 1  # 运行模型所需的 GPU 数量
--debug
```

如果一切正常，您应该看到屏幕上显示 “Starting inference process”：

```
[2024-01-12 18:23:55,076]
[opencompass.openicl.icl_inferencer.icl_gen_inferencer] [INFO] Starting
inference process...
```

评测完成后，将会看到：

```
dataset                                         version    metric         mode
     opencompass.models.huggingface.HuggingFace_model_repos_internlm-chat-7b
```

ceval-computer_network	db9ce2	accuracy	gen 31.58
ceval-operating_system	1c2571	accuracy	gen 36.84
ceval-computer_architecture	a74dad	accuracy	gen 28.57
ceval-college_programming	4ca32a	accuracy	gen 32.43
ceval-college_physics	963fa8	accuracy	gen 26.32
ceval-college_chemistry	e78857	accuracy	gen 16.67
ceval-advanced_mathematics	ce03e2	accuracy	gen 21.05
ceval-probability_and_statistics	65e812	accuracy	gen 38.89
ceval-discrete_mathematics	e894ae	accuracy	gen 18.75
ceval-electrical_engineer	ae42b9	accuracy	gen 35.14
ceval-metrology_engineer	ee34ea	accuracy	gen 50
ceval-high_school_mathematics	1dc5bf	accuracy	gen 22.22
ceval-high_school_physics	adf25f	accuracy	gen 31.58
ceval-high_school_chemistry	2ed27f	accuracy	gen 15.79
ceval-high_school_biology	8e2b9a	accuracy	gen 36.84
ceval-middle_school_mathematics	bee8d5	accuracy	gen 26.32
ceval-middle_school_biology	86817c	accuracy	gen 61.9
ceval-middle_school_physics	8accf6	accuracy	gen 63.16
ceval-middle_school_chemistry	167a15	accuracy	gen 60
ceval-veterinary_medicine	b4e08d	accuracy	gen 47.83
ceval-college_economics	f3f4e6	accuracy	gen 41.82
ceval-business_administration	c1614e	accuracy	gen 33.33
ceval-marxism	cf874c	accuracy	gen 68.42
ceval-mao_zedong_thought	51c7a4	accuracy	gen 70.83
ceval-education_science	591fee	accuracy	gen 58.62
ceval-teacher_qualification	4e4ced	accuracy	gen 70.45

ceval-high_school_politics	5c0de2	accuracy	gen	26.32
ceval-high_school_geography	865461	accuracy	gen	47.37
ceval-middle_school_politics	5be3e7	accuracy	gen	52.38
ceval-middle_school_geography	8a63be	accuracy	gen	58.33
ceval-modern_chinese_history	fc01af	accuracy	gen	73.91
ceval-ideological_and_moral_cultivation	a2aa4a	accuracy	gen	63.16
ceval-logic	f5b022	accuracy	gen	31.82
ceval-law	a110a1	accuracy	gen	25
ceval-chinese_language_and_literature	0f8b68	accuracy	gen	30.43
ceval-art_studies	2a1300	accuracy	gen	60.61
ceval-professional_tour_guide	4e673e	accuracy	gen	62.07
ceval-legal_professional	ce8787	accuracy	gen	39.13
ceval-high_school_chinese	315705	accuracy	gen	63.16
ceval-high_school_history	7eb30a	accuracy	gen	70
ceval-middle_school_history	48ab4a	accuracy	gen	59.09
ceval-civil_servant	87d061	accuracy	gen	53.19
ceval-sports_science	70f27b	accuracy	gen	52.63
ceval-plant_protection	8941f9	accuracy	gen	59.09
ceval-basic_medicine	c409d6	accuracy	gen	47.37
ceval-clinical_medicine	49e82d	accuracy	gen	40.91
ceval-urban_and_rural_planner	95b885	accuracy	gen	45.65
ceval-accountant	002837	accuracy	gen	26.53
ceval-fire_engineer	bc23f5	accuracy	gen	22.58
ceval-environmental_impact_assessment_engineer	c64e2d	accuracy	gen	64.52
ceval-tax_accountant	3a5e3c	accuracy	gen	34.69
ceval-physician	6e277d	accuracy	gen	40.82
ceval-stem	-	naive_average	gen	35.09

```
ceval-social-science                            -          naive_average  gen
                                                                      52.79
ceval-humanities                                -          naive_average  gen
                                                                      52.58
ceval-other                                     -          naive_average  gen
                                                                      44.36
ceval-hard                                      -          naive_average  gen
                                                                      23.91
ceval                                           -          naive_average  gen
                                                                      44.16
```

有关 `run.py` 支持的所有与 HuggingFace 相关的参数，请阅读 评测任务发起

除了通过命令行配置实验外，OpenCompass 还允许用户在配置文件中编写实验的完整配置，并通过 `run.py` 直接运行它。配置文件是以 Python 格式组织的，并且必须包括 `datasets` 和 `models` 字段。

示例测试配置在 configs/eval_demo.py 中。此配置通过 继承机制 引入所需的数据集和模型配置，并以所需格式组合 `datasets` 和 `models` 字段。

```python
from mmengine.config import read_base

with read_base():
    from .datasets.siqa.siqa_gen import siqa_datasets
    from .datasets.winograd.winograd_ppl import winograd_datasets
    from .models.opt.hf_opt_125m import opt125m
    from .models.opt.hf_opt_350m import opt350m

datasets = [*siqa_datasets, *winograd_datasets]
models = [opt125m, opt350m]
```

运行任务时，我们只需将配置文件的路径传递给 `run.py`:

```
python run.py configs/eval_demo.py
```

OpenCompass 提供了一系列预定义的模型配置，位于 `configs/models` 下。以下是与 opt-350m（`configs/models/opt/hf_opt_350m.py`）相关的配置片段:

```python
# 使用 `HuggingFaceCausalLM` 评估由 HuggingFace 的 `AutoModelForCausalLM` 支持的模型
from opencompass.models import HuggingFaceCausalLM

# OPT-350M
opt350m = dict(
       type=HuggingFaceCausalLM,
       # `HuggingFaceCausalLM` 的初始化参数
       path='facebook/opt-350m',
       tokenizer_path='facebook/opt-350m',
       tokenizer_kwargs=dict(
           padding_side='left',
           truncation_side='left',
           proxies=None,
           trust_remote_code=True),
```

```
        model_kwargs=dict(device_map='auto'),
        # 下面是所有模型的共同参数，不特定于 HuggingFaceCausalLM
        abbr='opt350m',                 # 结果显示的模型缩写
        max_seq_len=2048,               # 整个序列的最大长度
        max_out_len=100,                # 生成的最大 token 数
        batch_size=64,                  # 批量大小
        run_cfg=dict(num_gpus=1),       # 该模型所需的 GPU 数量
    )
```

使用配置时，我们可以通过命令行参数 `--models` 指定相关文件，或使用继承机制将模型配置导入到配置文件中的 `models` 列表中。

与模型类似，数据集的配置文件也提供在 `configs/datasets` 下。用户可以在命令行中使用 `--datasets`，或通过继承在配置文件中导入相关配置

下面是来自 `configs/eval_demo.py` 的与数据集相关的配置片段：

```
from mmengine.config import read_base  # 使用 mmengine.read_base() 读取基本配置

with read_base():
    # 直接从预设的数据集配置中读取所需的数据集配置
    from .datasets.winograd.winograd_ppl import winograd_datasets  # 读取 Winograd 配置，基于 PPL（困惑度）进行评估
    from .datasets.siqa.siqa_gen import siqa_datasets  # 读取 SIQA 配置，基于生成进行评估

datasets = [*siqa_datasets, *winograd_datasets]        # 最终的配置需要包含所需的评估数据集列表 'datasets'
```

数据集配置通常有两种类型：'ppl' 和 'gen'，分别指示使用的评估方法。其中 `ppl` 表示辨别性评估，`gen` 表示生成性评估。

此外，configs/datasets/collections 收录了各种数据集集合，方便进行综合评估。OpenCompass 通常使用 base_medium.py 进行全面的模型测试。要复制结果，只需导入该文件，例如：

```
python run.py --models hf_llama_7b --datasets base_medium
```

OpenCompass 通常假定运行环境网络是可用的。如果您遇到网络问题或希望在离线环境中运行 OpenCompass，请参阅 [FAQ - 网络 - Q1](FAQ - 网络 - Q1) 寻求解决方案。

可视化评估结果

评估完成后，评估结果表格将打印如下：

```
dataset    version    metric    mode      opt350m    opt125m
---------  ---------  --------  ------  ---------  ---------
siqa       e78df3     accuracy  gen         21.55      12.44
winograd   b6c7ed     accuracy  ppl         51.23      49.82
```

所有运行输出将定向到 `outputs/demo/` 目录，结构如下：

```
outputs/default/
├── 20200220_120000
├── 20230220_183030     # 每个实验一个文件夹
│   ├── configs         # 用于记录的已转储的配置文件。如果在同一个实验文件夹中重新运行了不同的实验，可能会保留多个配置
│   ├── logs            # 推理和评估阶段的日志文件
│   │   ├── eval
│   │   └── infer
│   ├── predictions   # 每个任务的推理结果
│   ├── results       # 每个任务的评估结果
│   └── summary       # 单个实验的汇总评估结果
├── ...
```

打印评测结果的过程可被进一步定制化，用于输出一些数据集的平均分 (例如 MMLU, C-Eval 等)。

关于评测结果输出的更多介绍可阅读 结果展示。

更多教程

想要更多了解 OpenCompass, 可以点击下列链接学习。

- https://opencompass.readthedocs.io/zh-cn/latest/