MELLM - an Automatic Evaluation Framework for LLM

Chengyu Cui, China
Li Zhu, China

Abstract

Recently, the landscape of large language models (LLMs) has been rapidly evolving, accompanied by a surge in new benchmarks and datasets designed to evaluate their performance. Despite many models claim to have surpassed ChatGPT on specific metrics and benchmarks, widespread skepticism persists. This skepticism arises from the recognition that evaluating LLMs is a customized matter, requiring a simplified yet reliable and unbiased evaluation process that can be easily accessed by everyone. To address this need, we introduce a novel benchmarking framework called Mutual-Evaluation-of-LLMs (MELLM). This framework revolutionizes the evaluation process by allowing multiple LLMs to serve both as examiners and examinees, synthesizing their results into a comprehensive total score. The key advantage of MELLM lies in its effortless extensibility, enabling the seamless integration of various LLMs as both examiners and examinees. Furthermore, we devise strategies to generate questions from seeds, ensuring a lightweight and equitable evaluation process. The results indicate that scores calculated by MELLM demonstrate strong linear correlation with both GPT-4 and human scoring. Codes and parts of questions and results are available at: https://github.com/dongrixinyu/JioNLP

1 Introduction

Large language models(LLMs) are evolving rapidly owing to their unprecedented capability in flexible and diverse applications, presenting challenges for the evaluation of LLMs. Many new benchmarks and metrics have been released aiming to evaluate LLMs reliably, unbiasedly and accurately. Nevertheless, more and more models are claiming to have surpassed ChatGPT[1], ranked among the top-N in LLM ranking leaderboard. It is still unclear for most people which LLM is better or which LLM is suitable to be used as a base model for fine-tuning. We analyse that the reasons for this phenomenon can be attributed to three main aspects:

(1) There are numerous evaluation benchmarks, such as MMLU[2], SuperGlue[3], C-EVal[4], OpenLLM[5], DynaBench[6], etc., along with evaluation metrics, such as BLEU[7], ROUGE[8], F1, win-rate and METEOR[9], etc., which assess the performance of models from various perspectives and angles. It proves that the evaluation of LLMs should be a customized matter, where different people focus on different aspect of capabilities. Besides, the abundance of evaluation criteria and metrics such as BLEU, ROUGE, etc., makes it challenging for non-AI professionals to understand and determine which indicator should be compared.

(2) Evaluated LLM may have seen public benchmarks and corpus in pre-training and finetuning process, leading to results that are over-estimated and untrustworthy. This proves that making new test samples unseen to evaluated LLM is key to unbiased evaluation results.

(3) The evaluation of LLMs is a highly time-consuming and labor-intensive manual task. It requires first collecting a certain number of test samples, then manually evaluating them and

summarizing the results. To address this, some benchmarks include multiple-choice questions with grounding truth. However these benchmarks have limitations compared to the actual needs of human. Some benchmarks provide tens of thousands of test samples for evaluation, thus executing it becoming costly. Besides, different evaluator may give different scores for the same answer, which can not be absolutely objective. This process leads to the vast majority of people being unable to easily evaluate the pros and cons of LLMs.

As a fact, In the realm of evaluation of LLMs, we assume that evaluation metrics should be concise and intuitive, the evaluation process should be straightforward with minimal cost, and the outcomes should be reliable and unbiased.

In this paper, we propose an easy-to-commence benchmarking framework to assess LLMs, Mutual-Evaluation-of-Large-Language-Models (MELLM), trying to alleviate labor cost, provide an reliable score by automatically execute the evaluation within several LLMs. We make three main studies:

- **Automatically generate, answer and grade customized questions from some seeds**. We assume that people only care about what they think important. Preparing customized testing questions automatically can avoid test-leakage, and focus on one specific aspect of capabilities. We arrange several LLMs both as examiners and examinees. Among LLMs, all questions are answered, all answers are graded by peers.
- **Mutual evaluation by several LLMs**. Since the number of scores for each question is large, we explore a novel algorithm to synthesize scores graded by LLMs examiners into a conprehensive total score. It is easy for Non-AI professionals to understand. We made experiments to prove the effectiveness of MELLM. The results indicate that scores calculated by MELLM demonstrate strong linear correlation with both GPT-4 and human scoring. Codes and parts of questions and results are available at JioNLP[1].

2 Related Work

2.1 Benchmarks

Along with the evolution of language models(LM), many benchmarks have been raised for evaluation. They have expanded from traditional natural language processing (NLP) tasks, such as text classification, sentiment analysis, and machine translation, to human-like exam questions and real users' needs.

This expansion is due to the impressive capabilities LLM have demonstrated in handling complex issues. Consequently, the benchmarks for assessing large language models now include a variety of types of tasks to comprehensively evaluate their performance.

For instance, [10] some benchmarks specifically test the model's reading comprehension ability, requiring the model to understand and answer questions from lengthy texts. [11], [12] examines the model's logical reasoning skills, asking the model to make reasonable inferences based on given premises; yet others test the model's ability to solve mathematical problems.

Besides, [13] some have noticed that public benchmarks might have been seen by LLMs to be evaluated and devised methods to generate customized questions.

2.2 Evaluation Metrics

1 https://github.com/dongrixinyu/JioNLP

The original evaluation method is manual grading by human, usually using Likert scale scoring or win-rate metric. This method is consuming highly significant manpower and subject to biases.

Several automatic evaluation metrics have been presented, such as F1 score, BM25, BLEU, ROUGE, and METEOR. Every metric mentioned above is derived from machine learning and NLP tasks, and is based on statistics of tokens and N-grams, namely lexicon-based. These metrics are no longer fit for evaluation of LLM due to its gap between tested results and actual performance of model.

Well-trained LLM can be used to tackle this problem. It can be divided into two main categories. One is reference-based, which means using a well-trained LLM to compute the semantic similarity between LLM generated result and the reference, such as FActScore[14], BERTScore[15]. These metrics outperform lexicon-based methods with more focus on contextual understanding. The other is reference-free[16], which means using a well-trained LLM to assess the generated result without comparing to a reference. For many NLP tasks, reference-free method exhibits high consistency with human assessment. But for human questions with complex logical reasoning, factual judgment and structured data output, this method still remains uncertain[17]. [18] inspect that ChatGPT achieves the new state-of-the-art correlations with human judgment on several benchmarks. However, on one hand, one examiner, either human or well-trained LLM, will inevitably bring bias to the evaluation. A group of human experts or LLMs will mitigate the bias by integration of results. On the other hand, empirically GPT-4[19] is recognized as the most capable model. [20] and [21] directly utilized GPT-4 as the examiner. But it still can not be trusted 100%. Moreover, if GPT-4 is not available or several LLMs are at a similar level to GPT-4, how should we deal with that?

Hence we present MELLM to solve this problem. We utilize multiple LLMs to evaluate each other, thus mitigating the biases. The results show that one LLM may infer which answer is better despite its incapability of giving correct answers.

3 Methodology

In this section, we discuss the methodology in of Mutual Evaluation of LLMs. The main steps is shown in Figure 1. It takes LLM(especially GPT-4) as an examiner to augment questions from human questions as seeds. Then, LLMs to be evaluated will be questioned with these augmented questions. We also use MMLU and customized datasets for comparing. Peer review will be executed and synthesized into a total score by EM algorithm. The scores by MELLM are scanned and compared to those by GPT-4 and human experts.

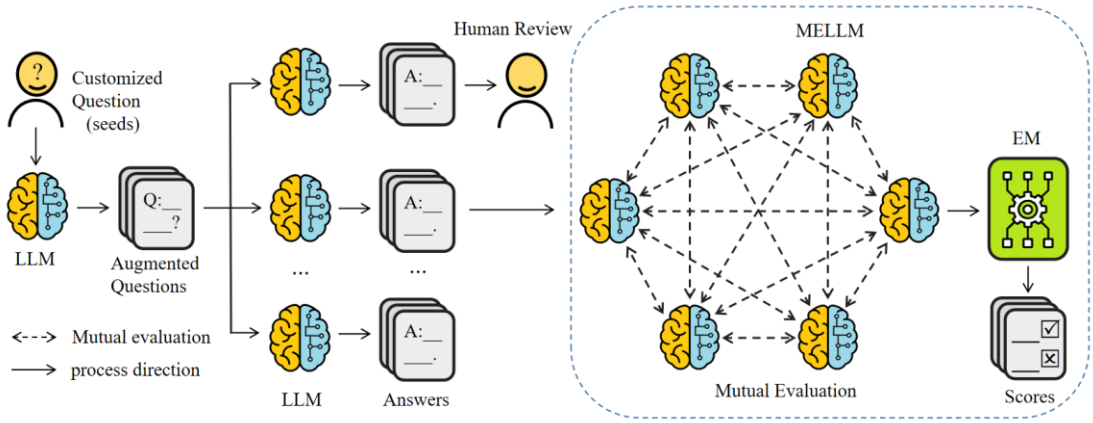

Figure 1: The process of MELLM

3.1 Dataset Construction

Our aim is to test the effectiveness of MELLM, rather than ranking LLMs upon several benchmarks. Besides, the evaluation of LLMs should be simple and accessible for everyone, not constrained in researchers. To facilitate this, we devised a method to automatically generate questions for evaluation from seeds.

The First step is to find seeds. We choose most popular keywords in WordStream[2] and Top 100 Most Googled Questions Globally[3] in 2023 as our seeds. Then apply these seeds in the prompt shown in Appendix A.1. When generating questions, we observed that although self-instruction is effective in generating questions for simple tasks and topics such as writing scripts, multiple-choice questions, it encounters difficulties in generating complex questions with long context and details. So we mainly put emphasis on generating simple questions. As for hard questions, we need to change some parts of seed questions, thus making diverse questions in different fields.

3.2 Evaluation Process

Suppose we have a total number of N LLMs to evaluate. In the previous session, we obtained a total of M test questions. By submitting these questions to different LLMs for answers, we receive a matrix consisting of $N \times M$ responses for evaluation.

Next, each response in the matrix of $N \times M$ responses is evaluated by the N LLMs respectively. We designed two prompts(shown in the Appendix A.2) to induce LLM to do grading. The difference of these two prompts is the appearance of reference.

Here we directly adopt a scoring system for each response as our metric. LLM should assign an appropriate score based on the quality of the answer provided. We believe that direct scoring is the most primitive and straightforward way to evaluate the quality of a model, which is closer to people's intuitive understanding of things, just like vocational qualification exams and admission exams in school. On the contrary, evaluation metrics such as ROUGE, BLEU, BERTScore, etc., have inherent biases with respect to the actual performance of LLM and are not easily understood by non-professionals. Moreover, the win-rate metric only evaluates the relative goodness of LLM and dismisses the absolute goodness of LLM for specific questions.

There are some other points to note. (1) We arrange each LLM to assess its own response to questions, thus comparing the scores to those graded by other LLMs. (2) The assessment of responses' quality by LLMs encompasses various dimensions, including the robustness, ethic, bias, trustworthiness, etc. This can be tuned in these two prompts by specifying dimensions in Appendix A.2. Here we just focus on the helpfulness of LLMs.

3.3 Synthesis of grading scores

We synthesize the grading scores by multiple LLMs. MELLM depends on several hypotheses: (1) If a model is of high quality and has strong capabilities, then its grading scores on other models' answers are likely to be more accurate, credible, and consistent. Conversely, a poor-quality model may produce evaluations that significantly deviate from actual results. (2) Every LLM has

2 https://www.wordstream.com/popular-keywords
3 https://www.semrush.com/blog/most-searched-keywords-google/

different level of ability to rating scores. Utilizing only one LLM can be biased and untrustworthy. Hence the quality of each response requires assessment by all N LLMs collectively. (3) While the scoring of a model's responses is determined collectively by all LLMs, it is significant to minimize the weight of scores from low-quality (or "garbage") LLMs and maximize the influence of scores from high-quality (or "excellent") LLMs. This approach helps ensure that the overall evaluation reflects the performance of the most reliable models, leading to a more accurate and trustworthy assessment of model outputs.

Here we use Expectation-Maximization(EM)[22] algorithm to synthesize all these scores into a total score for all LLMs. Denote:
i, j as the index of LLMs, m as the index of test questions,
$\omega_{i,m}$ as the weight of model i grading the answer of other models on question m,
ω_i as the weight of model i grading answers of other models,
$s_{i\to j,m}$ as the score of model i grading the answer of model j on question m.
$s_{j,m}$ as the weighted average score of model j on question m.
s_j as the total score model j. This score will be standardized to 100 points.
We can easily compute the average score of LLM j on question m and the final score of one LLM:

$$s_{j,m} = E(s_{j,m}) = \sum_i \omega_{i,m} s_{i\to j,m}$$

$$s_j = \sum_m s_{j,m}$$

Conversely, $\omega_{i,m}$ is determined by both the variance and final score of LLM i. ω_i is involved with $\omega_{i,m}$. Denote $\sigma_{j,m}^2$ as variation of LLM j on question m, σ_m^2 as the variation of question m, which depicts the difficulty level of a question.

$$\sigma_{j,m}^2 = E[(s_{i\to j,m} - s_{j,m})^2] = \frac{\sum_i (s_{i\to j,m} - s_{j,m})^2}{N}$$

$$\sigma_m^2 = \sum_j \sigma_{j,m}^2$$

$$\omega_{i,m} = \frac{\frac{\theta_1}{(s_{i\to j,m} - s_{j,m})^2} + \theta_2 (s_i - min(s_1, s_2, \dots, s_N)) + \omega_i}{Z_m}$$

$$\omega_i = \frac{\sum_m \sigma_m^2 \omega_{i,m}}{Z}$$

θ_1, θ_2 are hyper parameters tuning the impact of variation and scores respectively. Z_m denotes the sum of all unnormalized weight of all LLMs on question m, Z denotes the sum of all unnormalized weight of all LLMs. $s_i - min(s_1, s_2, \dots, s_N)$ means dismissing the impact of LLM with the worst capability, making the best LLM contribute to more weight on the final result. The three variables, ω_i, $\omega_{i,m}$ and s_j, are intercoupled and mutually influential. Upon given all the equations, we can first initialize a set of $\omega_i^{(0)}$ and $\omega_{i,m}^{(0)}$ as uniform distribution, which means all LLMs possess the same level of capability at the beginning, and then iterate over ω_i, $\omega_{i,m}$ and s_j until convergence by applying EM algorithm. The process of EM algorithm can be simplified as Figure 2.

Figure 2: variables dependencies when updating in EM algorithm

4 Experiments

Various companies and institutions are continuously advancing the development of LLMs. To demonstrate the effectiveness of MELLM in our paper, experiments are conducted using different LLMs at different times, on November 30, 2023, and June 6, 2024, respectively.

4.1 LLMs and Datasets

Experiment on 2023-11-30

We get access to a bunch of LLMs via either model weight files or Saas API interfaces, including ChatGPT, GPT-4, Claude[23], Gemini(named as Bard before)[24], Vicuna[25], Llama[26], Ernie[4], Qwen[5], Baichuan[6], 360AI[7]. These LLMs vary greatly in terms of parameter scale, capability, and training corpus. In this experiment, GPT-4 almost surpassed the performance of other LLMs in every aspect.

We applied these LLMs on three datasets, MMLU, customized questions and automatically generated questions by LLMs.

(1) MMLU. short for Multi-task Language Understanding, is a widely recognized and comprehensive benchmark for LLM that collects human exam questions in multiple-choice format across 57 tasks. We sampled 200 questions from the original datasets randomly. All questions are single-choice questions in English with ground-truth.

(2) Customized questions. MMLU probably has been seen by some LLMs when pretraining, thus causing testing leakage in evaluation. This applies to many wide-spread open benchmarks. We assume the evaluation of LLMs is a very personalized matter, with different individuals holding varying standards and scopes for the evaluation. Therefore, we have developed a customized test dataset that, while not large, covers topics such as world knowledge, natural sciences, social sciences, language, common sense, moral ethics and reasoning. This dataset comes from personal usage and can be accessed in JioNLP(codes in Appendix B.1).

(3) Generated questions by LLMs. When generating new questions by LLMs from seeds, we noticed the quality of generated questions vary greatly. Some LLMs can only raise short and simple questions concerning what, when, where, why, and can not created complex questions with longer context. Some generated results can not even be answered cause of confusion and factual inaccuracies. Here we only adopted GPT-4 to generate questions, and then reviewed and checked part of those questions manually. Obviously, the use of GPT-4 generating new questions retains the risk of testing leakage because GPT-4 may replicate what it has learned during pretraining. [13] has analyzed this problem. GPT-2[27] paper stated that the median 8-gram overlap rated between GPT-2's outputs and the exact completions from WebText test set articles were a mere 2.6%. We

4 https://yiyan.baidu.com/
5 https://github.com/QwenLM/Qwen
6 https://github.com/baichuan-inc/Baichuan2
7 https://ai.360.com/

infer it is the same to GPT-4. Nevertheless, in order to make the test unbiased, GPT-4 is excluded when performing MELLM on generated questions.

Experiment on 2024-06-06

We commence experiment again on June 6, 2024 to test the effectiveness of MELLM. We collect 22 LLMs(more than experiment on 2023-11-30) via API interfaces, including ChatGPT, GPT-4, Ernie, Qwen, Doubao[8], hunyuan[9], Moonshot[10], GLM[11], Yi[12], Mistral[13]. The LLMs of different versions from the above-mentioned companies have been tested and used, and the specific versions can be seen in the test results table.

The key difference between 2023-11-30 and 2024-06-06 is the performance of GPT3.5 and GPT-4. GPTs outperformed other LLMs on 2023-11-30, but have been caught up with by some competitors on 2024-06-06.

We applied LLMs on customized questions collected from daily use. This dataset has expanded and can also be accessed in JioNLP(codes in Appendix B.1). Most questions are in Chinese considering 21 out of 22 LLMs are developed by companies in China. Here, we need to declare that the MELLM algorithm is not limited to certain datasets; its effectiveness can be verified on any language, any field, and datasets of any difficulty.

4.2 Process of Grading Answers

Processes of Grading Answers by LLMs on both 2023-11-30 and 2024-06-06 are absolutely the same. When grading the answers by LLMs, we noticed interesting phenomenons.

(1) **Each LLM has its own personality**. Some LLMs are trained strictly to follow ethic rules of not expressing personal opinions or judgments. Hence the given scores always tend to be compromised. For example, a question is worth 1 point. The LLM always grades the answer with 0.5 points, declaring that it can not be responsible for its rating score. Conversely, some other LLMs tend to directly rate 1 or 0 points without hesitation.

Besides, some LLMs are very afraid of making mistakes, often refusing to answer questions or provide scores due to sensitive words. This is mainly caused by inadequate alignment operations during the model training process. For these occasions, MELLM can automatically detect the ineffectiveness and grade a low score.

(2) **Chain of Thought(CoT)** is effective in some cases. The scoring process for some responses should be done step by step to give accurate scores. In such cases, CoT can effectively guide to the correct score. However, GPT-4 tends to provide scores directly without a lengthy analysis process and still maintains a high level of accuracy.

(3) **Incapabilities**. Some LLMs can not output grading scores in json format, thus causing difficulty when parsing results into scores for calculation. We use GPT-4 to transform these responses into a json format(prompt shown in Appendix A.3), where GPT-4 and Doubao-128k acquired 100% accuracy. Some LLMs may refuse to answer questions or provide scores if encountering prohibited words. We believe the helpfulness outweigh moral constraint and assign

8 https://www.volcengine.com/product/doubao
9 https://cloud.tencent.com/act/pro/Hunyuan-promotion
10 https://kimi.moonshot.cn/chat/
11 https://chatglm.cn/
12 https://www.lingyiwanwu.com/
13 https://github.com/mistralai/mistral-inference

largest variable to these cases.

(4) **Length of response**. Some LLMs are prone to generate longer responses, whereas responses by GPTs are concise and clear. This may lead to differences in user payment and pricing strategies.

4.3 Analysis of MELLM results

Experiment on 2023-11-30

Experiment Setups: We commenced experiments by setting $\theta_1 = 0.5, \theta_2 = 0.02$ empirically. All the scores by MELLM may differ slightly according to different θ_1, θ_2. For different LLM, we utilize specific version of model noted in Table 1,2,3 and Figure 3. For all datasets, we tried both with and without GPT4 as an evaluator. Each test dataset has a different full score, so we standardize to 100 points uniformly. We have provided test scripts and some of customized datasets scores for use, which can be found in JioNLP.

Note that human experts' scores and GPT-4' scores are not the most convincing except MMLU, which contains grounding truth to refer to. They are probably close to the true scores than other LLMs. What we do is to compare MELLM to GPT-4 and human.

LLM	version	MMLU	MMLU (no gpt4)	GPT-4	Human
GPT-4	gpt-4	84.5	85.8	88.5	88.5
ChatGPT	gpt-3.5-turbo-1106	75.5	75.0	70.5	70.5
Vicuna	vicuna_13b	73.7	73.4	68.0	68.0
Claude	claude-2.0	80.0	80.8	78.5	78.5
Gemini	gemini-pro	76.2	76.9	72.5	72.5
llama	llama_2_70b	70.7	70.6	64.5	64.5
Ernie	ernie_bot_8k	71.0	71.5	65.0	65.0
Qwen	qwen_14b	66.4	66.6	57.5	57.5
Baichuan	Baichuan2_13b	61.4	61.8	48.0	48.0
360AI	360GPT_S2_V9	61.2	61.4	46.5	46.5

Table 1: Scores on 200 questions sampled from MMLU. We implement MELLM without GPT-4, which is labeled "(no gpt4)" in the headers.

LLM	version	Custom	Custom (no gpt4)	GPT-4	Human
GPT-4	gpt-4	89.7	92.8	89.8	90.6
ChatGPT	gpt-3.5-turbo-1106	81.4	83.6	76.9	78.8
Vicuna	vicuna_13b	76.8	76.0	66.1	65.2
Claude	claude-2.0	76.0	74.8	65.3	64.5
Gemini	gemini-pro	80.2	82.3	74.7	75.8
llama	llama_2_70b	70.7	71.9	55.1	53.2
Ernie	ernie_bot_8k	83.0	85.5	78.1	78.9
Qwen	qwen_14b	82.4	83.1	76.6	77.0
Baichuan	Baichuan2_13b	71.4	71.5	60.6	58.9
360AI	360GPT_S2_V9	71.2	71.9	58.9	55.7

Table 2: Scores on customized questions.

LLM	version	Generate	Generate (no gpt4)	GPT-4	Human
GPT-4	gpt-4	91.2	92.8	88.8	89.6
ChatGPT	gpt-3.5-turbo-1106	82.4	83.7	76.2	77.5
Vicuna	vicuna_13b	78.1	76.3	65.1	64.2
Claude	claude-2.0	75.0	75.5	65.0	63.5
Gemini	gemini-pro	81.2	79.7	73.7	71.8
llama	llama_2_70b	68.3	71.1	52.1	50.2
Ernie	ernie_bot_8k	84.0	84.2	77.3	76.7
Qwen	qwen_14b	81.6	81.3	75.6	75.0
Baichuan	Baichuan2_13b	72.4	73.2	57.6	58.9
360AI	360GPT_S2_V9	72.3	73.0	55.9	56.7

Table 3: Scores on generated questions.

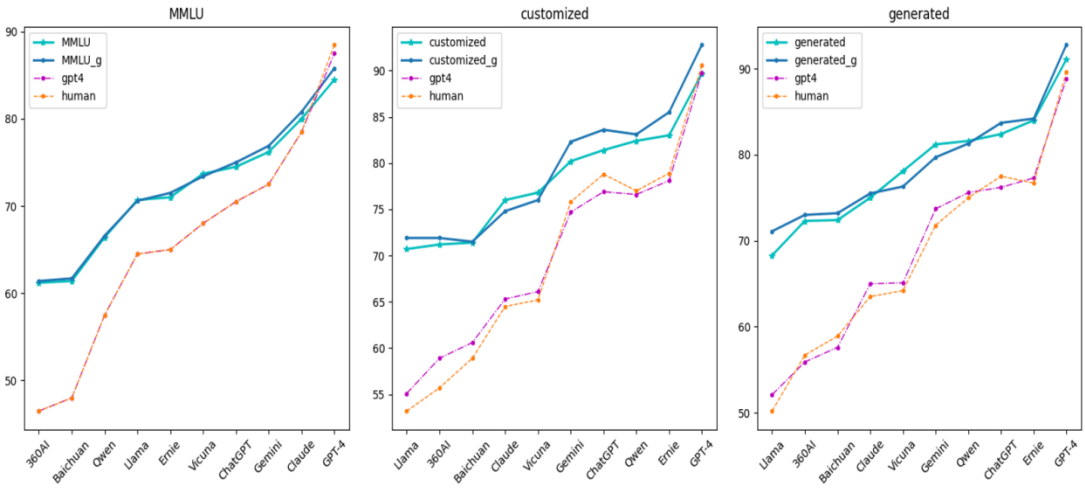

Figure 3: All scores evaluated by MELLM, MELLM without GPT-4, GPT-4 and human. "_g" means applying MELLM without GPT-4.

Here we noticed that there is a gap between scores calculated by MELLM and graded directly by GPT-4 and human. This is caused by the weighted average of all LLMs. This gap does not indicate that the MELLM algorithm is ineffective. What we should be concerned about is the linear correlation between the results obtained by the MELLM and those obtained by GPT-4 and human. We choose Pearson correlation coefficient as the metric. The closer this indicator is to 1, the better the effectiveness of the MELLM. The details is shown in Table 4.

Pearson coefficient	GPT4	human
MELLM-MMLU	0.998	0.998
MELLM-MMLU(no gpt4)	0.997	0.997
MELLM-customized	0.992	0.989
MELLM-customized(no gpt4)	0.986	0.984
MELLM-generalized	0.989	0.990
MELLM-generalized(no gpt4)	0.975	0.982

Table 4: Pearson correlation coefficient between MELLM and GPT-4, human experts

We noticed some phenomenons about the results:

- Every LLM except GPT-4 gives GPT-4's answers higher score than itself. This indicates that one LLM may infer which answer is better despite of the incapability to provide correct answer. This ensures that applying MELLM without GPT-4 still works well.
- MMLU exhibits the test leaderboard[14] on the whole dataset. The pearson coefficient between 200 sampled questions and the whole datasets of MMLUE is 0.9562. It means that the evaluation of LLMs can achieve a relatively correct result based on a small dataset.
- Providing grounding truth or not makes difference on MELLM. GPT-4 offered 100% percentage of correctness when grading scores on MMLU given grounding truth. Besides, we noticed that LLMs except GPT-4 provide scores in high linear correlation with the grounding truth. In contrary, some of customized questions and generated questions are lack of standard correct answers, which cause LLMs works imperfectly and lower Pearson value.
- LLMs shows difference on each dataset. Llama works badly on Chinese. It always generates English response given Chinese prompt. Vicuna outperforms llama due to further fine-tuning. Most LLMs released by Chinese (such as Ernie, Qwen, Baichuan and 360AI) perform well on customized datasets(questions mainly in Chinese) and badly on MMLU(questions in English).
- Scaling law is evaluated. Most LLMs with 7B, 13B params perform badly and are lack of logic in both answering and grading in contrast with those with large scale.

Experiment on 2024-06-06

Experiment Setups: We set $\theta_1 = 0.5, \theta_2 = 0.02$ empirically. For the expanded customized datasets, we tried the grading result by pure GPT4 and MELLM. When applying MELLM, we adopted 10 LLMs as evaluators, including *Doubao-lite-4k-character-240515, Doubao-pro-128k-240515, ERNIE-3.5-8K, ERNIE-4.0-8K-Preview-0518, gpt-3.5-turbo-16k, gpt-4-turbo-2024-04-09, hunyuan-standard-256K, Moonshot-v1-32k-v1, qwen1.5-14b-chat, qwen2-72b-instruct*. Due to the considerable effort required for manual scoring, this experiment did not involve human scoring participation. The result is shown in Table 5.

LLM version	gpt-4 score	MELLM score	gpt-4 rank	MELLM rank
ERNIE-4.0-8K-Preview-0518	84.5	85.7	4	1
ERNIE-3.5-8K	85.4	85.5	1	2
Doubao-pro-128k-240515	84.9	84.4	3	3
qwen2-72b-instruct	84.4	84.2	5	4
gpt-4-turbo-2024-04-09	85.2	83.0	2	5
hunyuan-pro	83.5	82.8	6	6
Doubao-pro-4k-browsing-240524	82.1	82.7	8	7
Moonshot-v1-32k-v1	82.7	81.9	7	8
GLM3-130B-v1.0	81.5	80.5	9	9
Moonshot-v1-8k-v1	79.5	80.4	12	10

14 https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu

Moonshot-v1-128k-v1	81.0	80.3	11	11
qwen-plus	81.5	80.3	10	12
Yi-34B-Chat	74.5	76.0	14	13
qwen1.5-14b-chat	77.6	75.9	13	14
hunyuan-standard-256K	73.3	73.9	15	15
hunyuan-standard	70.1	72.1	16	16
gpt-3.5-turbo-16k	65.7	70.2	17	17
Doubao-lite-128k-240428	64.2	69.2	19	18
Doubao-lite-4k-character-240515	61.2	67.6	20	19
hunyuan-lite	65.6	66.2	18	20
Mistral-7B-instruct-v0.2	58.4	61.6	21	21
qwen1.5-110b-chat	46.8	49.1	22	22

Table 5: Scores by GPT-4 and MELLM, sorted in descending order based on the MELLM algorithm. All details concerning the experiments can be acquired in JioNLP.

From this table, we can infer that the scoring results from GPT-4 are roughly consistent with MELLM, which integrates the scoring results from 10 models illustrated above. In the MELLM algorithm, the weight of the GPT-4 scoring is only 12%. The model with the highest scoring weight is ERNIE 3.5, at 21%. The Pearson coefficient between GPT-4 and MELLM is 0.987. Therefore, it can be seen that the MELLM model is effective despite lower influence of GPT-4.

We noticed some phenomenons about the results:

- GPT-4 assigned a higher score to Ernie3.5 over Ernie4, which is probably inappropriate. In contrast, MELLM calculated a more suitable score, Ernie4 is empirically better than Ernie3.5.
- Both Ernie and Qwen are tested on 2023-11-30 and 2024-06-06. The gap between them and GPT-4 have narrowed. Both open-source LLMs, such as Qwen2, and closed-source LLMs, such as Ernie and Doubao are evolving rapidly.
- Compared to experiment on 2023-11-30, by 2024-06-06, the GPT-4 model is no longer the best model on some datasets, nor can it significantly outperform other models. This demonstrates the rapid progress of other LLMs. Empirically, we are more inclined to trust the GPT-4 model to evaluate LLMs more effectively. However, experiments show that even without the participation of GPT-4, or when GPT-4 is no longer significantly ahead of other models, the MELLM algorithm remains more effective.
- For every company, such as OpenAI(GPT), Baidu(Ernie), Alibaba(Qwen), Bytedance(Doubao), Tencent(hunyuan), Moonshot, LLM-pro versions got better scores than its basic versions. *qwen1.5-110b-chat* is an exception despite its 110 billion parameters, due to its extreme sensitivity to sensitive words.

5 Conclusion

In this paper, we propose Mutual Evaluation of LLMs(MELLM), which aims to automatically, reliably evaluate LLMs without human experts. This approach serves to mitigate the biases of only one evaluator and synthesize the result as a total score. We sampled MMLU benchmark, construct customized dataset, and generate new questions to test the effectiveness. We hope that MELLM would become a reliable method to be trusted and used by public.

In the future, as AIGC (AI-Generated Content) evolves into the realm of multi-modality, encompassing areas such as images, videos, and audio, the evaluation of models will also progress accordingly. MELLM, essentially an automated model evaluation based on the linguistic modality, implies that as long as multi-modal AI models possess a linguistic modality, the MELLM algorithm can be applied to complete model evaluation. Therefore, MELLM is a highly extensible evaluation algorithm with strong vitality.

6 Limitation

Our proposed approach is with the following shortcomings and needs further investigation:

(1) **Hard questions**. This approach is based on the hypothesis that more than half of LLMs to be evaluated have a good understanding on world knowledge and the tested benchmarks. What if we provide LLMs with extremely hard questions that even the most capable LLM such as GPT-4 can not answer correctly? It means that there is no one LLM can generate correct answer, thus MELLM can not automatically choose the correct LLM as the evaluator.

(2) **Ethical Consideration**. We dismiss moral constraint of LLM and focus on the effectiveness of solving problems. LLMs may differ greatly and hold opposite opinions on the ethical consideration, thus grading diverse scores. This could potentially lead MELLM to fail to converge when applying EM like *qwen1.5-110b-chat* in the experiment on 2024-06-06.

Reference

[1] OpenAI, Introducing chatgpt, 2022.

[2] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 (2020).

[3] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems 32 (2019).

[4] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. arXiv preprint arXiv:2305.08322 (2023).

[5] HuggingFace. 2023. Open-source Large Language Models Leaderboard. https://huggingface.co/spaces/ HuggingFaceH4/open_llm_leaderboard.

[6] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in NLP. arXiv preprint arXiv:2104.14337 (2021).

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311 – 318, 2002.

[8] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in Text summarization branches out, pp. 74 – 81, 2004.

[9] S. Banerjee and A. Lavie, Meteor: An automatic metric for mt evaluation with improved

correlation with human judgments, in Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65–72, 2005.

[10] Liang Xu, Anqi Li. 2023 SuperCLUE: A Comprehensive Chinese Large Language Model Benchmark.arXiv:2307.15020 (2023)

[11] Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A Benchmark for Tool-Augmented LLMs. arXiv:2304.08244 [cs.CL]

[12] Qiantong Xu, Fenglu Hong. 2023 On the Tool Manipulation Capability of Open-source Large Language Models. arXiv:2305.16504 2023

[13] Yushi Bai, Jiahao Ying. 2023 Benchmarking Foundation Models with Language-Model-as-an-Examiner. arXiv preprint arXiv:2306.04181 (2023).

[14] 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. arXiv preprint arXiv:2305.14251 (2023)

[15] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019).

[16] Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. arXiv preprint arXiv:2304.00723 (2023).

[17] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, Survey of hallucination in natural language generation, ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.

[18] J. Wang, Y. Liang, F. Meng, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou, Is chatgpt a good nlg evaluator? a preliminary study, arXiv preprint arXiv:2303.04048, 2023.

[19] OpenAI, Openai: Gpt-4, 2023.

[20] J. Fu, S.-K. Ng, Z. Jiang, and P. Liu, Gptscore: Evaluate as you desire, arXiv preprint arXiv:2302.04166, 2023.

[21] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, Gpteval: Nlg evaluation using gpt-4 with better human alignment, arXiv preprint arXiv:2303.16634, 2023.

[22] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the royal statistical society. Series B (methodological), 1977: 1-38.

[23] Anthropic, Anthropic: Claude, 2023.

[24] Google, Google: Gemini, 2023.

[25] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, 2023 Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., 2023 Llama: Open and efficient foundation language models, arXiv:2302.13971, 2023.

[27] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

Appendix

A Example prompts

A.1 Question generation

Generate questions on topic seeds.

Please generate a question and its correct answer concerning {topic seed}. Note:
- The question should be in multiple choice format.
- The answer should be 100% correct.

Generate questions on questions seeds. When applying on customized questions, we translate the prompt into Chinese.

I will give you a sample question and the correct answer.
【Question】:
```
{question}
```
【Correct Answer】:
```
{correct_answer}
```
Please give me a similar question and its correct answer. Note: the topic and question format should be in consistent. And the answer should be 100% correct.

A.2 Grading generation

Grading generation prompt with reference:

I will give you a question and a corresponding answer which is provided by a person.
Please give me a score measuring if this answer is correct and its quality.
【Question】:
```
{question}
```
【Correct Answer】:
```
{correct_answer}
```
【Answer of this person】:
```
{response}
```
According to the above, please give me a score measuring if this answer is correct and its quality. The highest score is {score}, grading granularity is 0.5:

Grading generation prompt without reference:

I will give you a question and a corresponding answer which is provided by a person.
Please give me a score measuring if this answer is correct and its quality.
【Question】:
```
{question}
```
【Answer of this person】:
```
{response}
```
According to the above, please give me a score measuring if this answer is correct and its quality. The highest score is {score}, grading granularity is 0.5:

A.3 Normalization of Score

I will give you a piece of text which is an evaluation by an evaluator on a test taker's answer:
```
{grading_result}
```
The full score for this question is {score} points. Based on the above evaluation, please tell me how many points the evaluator gave to the result?
Note:
- If the text does not explicitly state the score, then tell me what score the evaluator is most likely to have given.
- If the content of the evaluation does not match the score given, such as giving a high score for a wrong answer or a highest score despite flaws, then tell me the correct score based on the content of the evaluation.
- If the evaluation refuses to rate or the scoring preamble is inconsistent with the postscript and the logic is poor, then score -1 point.
- Please tell me the score in JSON format, with the only field name being `score`, and do not return redundant information except JSON.

B Trivials

B.1 Codes to loading customized datasets.

This codes in written in Python and should be run before installation by “pip install jionlp”, where the github link is https://github.com/dongrixinyu/JioNLP

```
import jionlp as jio

llm_test = jio.llm_test_dataset_loader(version='1.1')
print(llm_test[15])
llm_test = jio.llm_test_dataset_loader(field='math')
print(llm_test[5])
```