InstructEval: Instruction-Tuned Text Evaluator from Human Preference

Anonymous ACL submission

Abstract

This paper explores to construct a general text evaluator based on open-source Large Language Models (LLMs), a domain predominantly occupied by commercial counterparts such as GPT-4. Recognizing the limitations of open-source models like Llama in evaluative tasks, we introduce InstructEval, a general multi-aspect text evaluator developed through instruction tuning of open-source LLMs. To overcome the shortage of annotated resources for multi-aspect evaluations, InstructEval combines extensive open Human Preference Modeling (HPM) datasets with a small set of multi-aspect annotated data. This approach not only enhances effectiveness in overall evaluation tasks but also exhibits improved performance in multi-aspect evaluation tasks. As demonstrated by our extensive experiments, InstructEval achieves comparable or superior performance to commercial LLMs like ChatGPT or GPT-4 in terms of both overall and multi-aspect evaluation. Our model and datasets will be open released to the community.

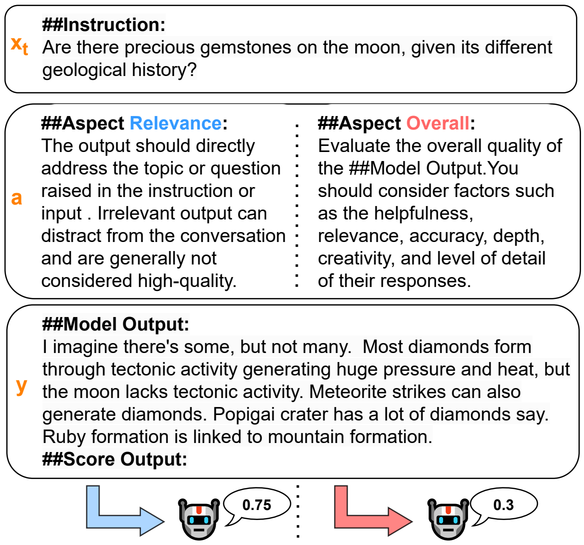

Figure 1: A sample of InstructEval evaluating a piece of text y from task x_t based on a specific aspect (left) or an overall aspect (right).

1 Introduction

Recent advancements in LLMs, exemplified by renowned models like ChatGPT, have showcased their impressive zero-shot capabilities in a generative manner, empowering them to effectively handle a wide range of arbitrary human instructions (Brown et al., 2020; Wei et al., 2022). Nonetheless, assessing the text quality presents a significant challenge due to the complexity of tasks and the necessity for multi-aspect evaluation (Ethayarajh and Jurafsky, 2021; Chang et al., 2023).

In recent research, leveraging the generalization capabilities of Large Language Models (LLMs) as evaluators for text generation tasks has gained popularity. For example, a popular method involves using pairwise comparisons, with models like GPT-4 serving as benchmarks for identifying superior samples (Zheng et al., 2023). However, the use of commercial LLMs, such as GPT-4, poses limitations due to their high cost and potentially slow and inconsistent response times, which impede their practical application in large-scale evaluations. In contrast, while open-source LLMs present a more accessible and efficient alternative, they tend to be less effective as evaluators, especially for multi-aspect evaluation. For instance, in Table 3, we observed that without further training, Llama (Touvron et al., 2023) based GPTScore fails in effectively assessing dialog-level data.

In this paper, we aim at investigating the construction of a general multi-aspect text evaluator based on open-source LLMs. Constructing such an evaluator is challenging mainly due to the following two reasons: (1) Limited resources: Training a text evaluator is hindered by the scarcity of annotated samples. (2) Complexity of evaluation instructions: An effective text evaluator is required to comprehend intricate task instructions and assess text samples across multiple aspects. To address these challenges, we propose to fine-tune LLMs using publicly available Human Preference Modeling (HPM) datasets. These datasets are designed to capture human preferences regarding responses to various questions and instructions. Despite their human-assessed nature, HPM datasets represent valuable labeled resources, offering a diverse and rich set of instructions, as outlined in Table 1. These diverse and rich instructions empower language models not only to evaluate the overall quality of the target task but also to extend their capability for assessing various aspects and instructions in a zero-shot manner (Chung et al., 2022; Longpre et al., 2023).

To put this insight into practice, we propose InstructEval, a multi-aspect text evaluator that leverages instruction tuning on human preference datasets. Initially, we establish a standardized prompt format, termed "eval-instruct", designed to unify Human Preference Modeling (HPM) data from diverse sources. This format is composed of three integral parts: the task definition, the aspect definition, and the target sample. Such a structure allows for the clear delineation and decomposition of both explicit and implicit evaluation criteria within the HPM data. Subsequently, we engage in the fine-tuning of LLMs using the entirety of available open-source HPM data structured in the eval-instruct format. This process substantially enhances the LLMs' capabilities in assessing textual quality comprehensively. To further refine the LLMs' proficiency in complex multi-aspect evaluations, we construct a small set of multi-aspect eval-instruct for augmentation. This set is generated through a methodical process of random sampling, combining various task instructions and aspect definitions, and employing GPT-4 for the creation of specific assessment preferences. Through joint fine-tuning of LLMs using both the publicly available HPM data and our constructed multi-aspect labeled data, InstructEval not only demonstrates improved efficacy in evaluating the overall quality of text samples but also exhibits enhanced performance in multi-aspect evaluation tasks.

Extensive experimentation in both overall helpfulness and multi-aspect evaluation demonstrate that our proposed method, InstructEval, achieves comparable or even superior performance to ChatGPT or GPT-4, despite fine-tuning with only 7B or 13B parameters of Llama. Furthermore, we conduct a thorough analysis to examine the contributions of each resource in enhancing the evaluation capabilities of InstructEval. This analysis allows us to gain valuable insights into the specific benefits provided by each resource. Our contributions can be summarized into three folds:

- We propose a novel instruction-tuning method to make full use of human preference modeling datasets in text evaluation. The model and dataset will be released to the community.
- A general and multi-aspect text evaluator InstructEval is implemented and achieves comparable or superior performance to commercial LLMs like ChatGPT or GPT-4.
- Our experiments reveal the relations between overall human preference and multi-aspect evaluations. Additionally, we provide a thorough analysis on how current available HPM resources benefit text evaluation.

2 Preliminaries and Related Works

Our work is closely related to the three domains discussed below, and there are subtle connections among these three domains as well.

Instruction tuning Instruction tuning is a technique employed to assist LLM in better comprehend and respond to a wide range of diverse instructions provided by humans (Brown et al., 2020). Formally, with each task $t \in \mathcal{T}$ written in explicit instruction (may also include a task input), instruction tuning trains a model by maximizing:

$$\mathop{\mathbb{E}}_{t \sim \mathcal{T}, \boldsymbol{y} \sim t} \left[p(\boldsymbol{y}|t) \right], \tag{1}$$

where $\boldsymbol{y}$ is either written by human, such as Flan (Longpre et al., 2023), Supernatrual (Wang et al., 2022), etc., or generated by other LLMs, such as Alpaca (Taori et al., 2023).

Human Preference Modeling (HPM) HPM, often referred to as Reward Modeling (RM) in the literature on Reinforced Learning from Human Feedback (RLHF), leverages human preference samples to train an evaluation model (Christiano et al., 2017; Stiennon et al., 2020). The main goal of HPM is to ensure that the evaluation model aligns with human judgment and accurately assesses the quality or performance of given samples. HPM is usually performed in a pairwise manner: given a pair of output samples $(\boldsymbol{y}_w, \boldsymbol{y}_l)$, where $\boldsymbol{y}_w$ is the sample human preferred over $\boldsymbol{y_l}$, HPM trains the evaluation model by maximizing:

$$\mathop{\mathbb{E}}_{t\sim\mathcal{T},(\boldsymbol{y_w},\boldsymbol{y_l})\sim t}\left[p(\boldsymbol{y_w}\succ\boldsymbol{y_l}|t)\right], \qquad (2)$$

where $p(\boldsymbol{y_w}\succ\boldsymbol{y_l}|t)$ denote the probability of evaluation model select $\boldsymbol{y_w}$ over $\boldsymbol{y_l}$. Current HPM in text usually involves finetuning multiple domains and instructions (Ouyang et al., 2022).

Text Evaluation Models Previous text evaluation mainly focus on assessing a single task (Kryscinski et al., 2020) or a set of tasks in generation (Zhang et al., 2020; Wu et al., 2023). Benefiting from the powerful generalization ability of LLMs, current text evaluation models have the capacity to assess any given task. An evaluation model M should evaluate the quality of sample $\boldsymbol{y}$ based on task requirement t and the required evaluation aspect a_i by outputting a score $s_{ta} = M(\boldsymbol{y}|t, a_i)$. Recent works focus on how to prompting general LLMs for text evaluation. GPTScore (Fu et al., 2023) and G-EVAL (Liu et al., 2023) study how to better use GPTs for multi-aspect text evaluation. Zheng et al. (2023) investigate the possibility of using GPT-4 as a replacement for human experts as evaluators and discovered its positional biases.

3 InstructEval

In our proposed InstructEval, our target is to construct a LLM with the ability to evaluate not only from overall quality but also based on specific aspects defined. LLMs should provide objective evaluations even in situation where a particular task and aspect that never appeared in training set.

To achieve this objective, our initial step proposes a standardized prompt format called "eval-instruct", which explicitly defines evaluation task and aspects. Subsequently, we finetune a base LLM using HPM samples to equip the LLM with the capability to evaluate the overall aspect of the given task. In order to further augment the LLM's capacity for specific aspect assessments, we create an additional set of samples evaluated in diverse aspects. These additional samples are then added to our training set, contributing to the training of our final InstructEval. Intuitively, the design of an InstructEval can be deconstructed into three fundamental components: **input format**, **scoring format** and **instruction tuning**. In the subsequent section, we will provide a detailed introduction to each part.

3.1 Input Format: Eval-Instruct

Previous approaches have employed either point-wise evaluation prompts, which involves a single target sample $\boldsymbol{y}$ in its input, or pairwise evaluation prompts, which involves a pair of target samples. In InstructEval, we choose to employ a point-wise prompt instead of a pairwise prompt, and this choice is motivated by the following reasons: 1. Flexibility in inference: Many application and evaluation scenarios require list-wise ranking, which can be computationally expensive with pairwise estimation due to the $\mathcal{O}(n^2)$ number of comparisons involved. 2. Memory cost in training: When using pairwise comparisons, including an additional variable $\boldsymbol{y}$ in the prompts significantly increases their length. This can present memory challenges during the fine-tuning LLMs. For point-wise prompt, it is essential to include three essential components:

1. Instruction $\boldsymbol{x}_t$: This defines the requirement of task t in text.

2. Evaluation aspect $\boldsymbol{a}$: This defines and explains the required evaluation aspect in textual form.

3. Target sample $\boldsymbol{y}$: This represents the textual output that needs to be evaluated.

Based on the above three components, we propose a standard point-wise input format, called "eval-instruct", for our instruction tuning. This format composes the required components by concatenating them as follows:

$$\boldsymbol{e}_{ta} = \boldsymbol{x}_t \oplus \boldsymbol{a} \oplus \boldsymbol{y}. \qquad (3)$$

In this format, the requirements for evaluation, $\boldsymbol{x}_t$ and $\boldsymbol{a}$, are both defined in natural language. The concatenations $\oplus$ are performed using a specific template. As illustrated in Figure 1, the eval-instruct includes special tokens: "##Instruction" placed before $\boldsymbol{x}_t$ to indicate the start of the instruction. "##Aspect {aspect name}:{aspect definition}" is used to format $\boldsymbol{a}$ and its definition. "##Model Output" and "##Score Output" respectively indicate start of $\boldsymbol{y}$ and score output.

3.2 Scoring Format

Given an eval-instruct as input, the evaluation model first generates its last hidden state $\boldsymbol{h} \in \mathcal{R}^n$. We employ and compare two different methods of scoring based on $\boldsymbol{h}$, namely regression score and expected Likert score.

Regression Score applies a newly initialized linear head $\boldsymbol{W} \in \mathcal{R}^n$ on the top of pre-trained LLM and output the final logits score by $s = \boldsymbol{W}^T \boldsymbol{h}$, This scoring format has been widely used in the reward models of recent works in RLHF (Ouyang et al., 2022; Touvron et al., 2023). However, this format shifts the model from generating tokens to classifying representations, which introduce inconsistencies between the pre-training and finetuning.

Expected Likert Score (ELS) Instead introducing new parameters, ELS generates number tokens (0-9) from the word list. To ensure efficient gradient back-propagation during training, we use the expected score of generating numbers:

$$s_{ta} = \sum_{w \in V_n} w * p(w|\boldsymbol{e}_{ta}) / \sum_{w \in V_n} p(w|\boldsymbol{e}_{ta}), \quad (4)$$

where $V_n = \{0, \ldots, 9\}$ is the list of number tokens in the word list, $p(w|\boldsymbol{e}_{ta})$ denotes the likelihood of generating token w. The denominator of this equation normalizes the probability distribution over $w \sim V_n$. The benefit of using ELS is that it maintains the same generation paradigm as in the pre-training phase. However, it is also important to consider that ELS constrains the score within the range of [0, 9]. This limitation may restrict the expressive range of the scores, potentially limiting the evaluator's ability to provide nuanced assessments. We further compare with these two scoring formats through experiments.

3.3 Instruction Tuning With Aspects

We now delve into explaining the training process of the evaluator. Following HPM, given a pair of eval-instruct (e^w_{ta}, e^l_{ta}) for task t, where e^w_{ta} is preferred by human than e^l_{ta} under the aspect a, we first calculate their evaluated score by model as (s^w_{ta}, s^l_{ta}) and train the model by:

$$\mathcal{L} = - \sum_{t \sim \mathcal{T}, a \in \mathcal{A}} \log \sigma(s^w_{ta} - s^l_{ta}), \quad (5)$$

where $\mathcal{T}$ and $\mathcal{A}$ are the set of ranking tasks and aspects, σ is the sigmoid function.

The remaining challenge lies in collecting a sufficient number of eval-instructs that are evaluated under diverse aspects. We apply two types of data to reach this goal: HPM datasets and our constructed Multi-aspect eval-instruct data.

HPM Data with Overall Aspect Reported in first block in Table 1, publicly available HPM datasets offer a substantial number of samples on overall aspects evaluation. As a result, fine-tuning with these datasets leads to an evaluation model that closely aligns with human preferences, enabling comprehensive evaluations of the samples. In practice, as illustrated on the right side of Figure 1, we set the aspect name of these samples to "Overall" and provide a comprehensive aspect definition.

Multi-aspect Eval-instruct Construction We construct an addition multi-aspect eval-instruct to ensure the ability of multi-aspect evaluation. Although the possibilities for task instructions are infinite, commonly used evaluation aspects are often limited and shared across different tasks. For instance, summary evaluation commonly involves 4 aspects (Fabbri et al., 2021; Zhang et al., 2019), while dialogue evaluation employs 7 aspects (Mehri and Eskenazi, 2020), with coherence, consistency, and relevance being shared aspects between them.

Leveraging this characteristic, we first list out a full aspect list involving all the commonly used evaluation aspects and their corresponding definitions. For convenience and flexibility, we prompt ChatGPT to generate this full aspect list, as it has demonstrated great proficiency in generating a wide array of aspects and their definitions. Next, for a given HPM dataset (first block of Table 1) with a task type t, we carefully select a candidate aspect set $\mathcal{A}_t$ from the previously generated full aspect list. This selection process ensures that the chosen aspects are relevant and appropriate for evaluating the task t defined in the dataset. Subsequently, we randomly sample several data points from task t and assign them with an aspect randomly selected from the aspect set $\mathcal{A}_t$. In the final step, we utilize GPT-4 (OpenAI, 2023) to relabel these samples with various assigned aspect.

We select 500 samples from each dataset, excluding Harmful-HH, and filter out some samples to ensure label balance. Additionally, we incorporate human-annotated, multi-aspect data from OpenAI-Sum, referred to as "OpenAI-Sum aspect". The detailed statistics of this multi-aspect eval-instruct data are presented in the second block of Table 1. We combine these data with HPM dataset, reformulating them into eval-instruct format for fine-tuning. For further details on this process, please refer to Appendix 7.1.

Datasets	Labeler	Task Types	Aspect	Comparisons
SHP	Human	Dialogue	Overall Helpfulness	348,718
WebGPT	Human	QA	Overall helpfulness	19,578
OpenAI-Sum	Human	Summary	Overall quality of summary	92,858
Helpful-HH	Human	Dialogue	Overall Helpfulness	118,266
Harmful-HH	Human	Dialogue	Overall Saftyness	42,538
Instruct-GPTJ	Human	QA	Overall helpfulness	33,143
Openai-Sum aspect	Human	Summary	Specific Aspects	68,469
Multi-aspect instruction	GPT-4	Mixture above	Specific Aspects	3,500

Table 1: Public HPM datasets and multi-aspect labeled dataset used to train InstructEval.

4 Training and Experiment Setup

4.1 Training Settings

Training Datasets The overall statistics of pre-processed training datasets are listed in Table 1. The open-sourced HPM datasets are Stanford Human Preference (SHP), WebGPT, OpenAI-Sum, Helpful-HH and Harmful-HH, Instruct-GPTJ, and an additional Openai-Sum aspect. More details of our training data is provided in Appendix 7.2.

Model Settings We select Llama v2, current state-of-the-art open-source pretrained language model, as the base model for fine-tuning. Two versions of the model are trained, one with 7 B parameters and a larger variant with 13 B parameters. Detailed hyperparameters for finetuning are provided in Appendix 7.3.

4.2 Experiment Settings

We conducted validation experiments on two main categories of evaluation tasks. Notice: *We have conducted a careful examination to ensure that there are no instances of overlapping samples between our training data and the testing benchmark.*[1]

Multi-aspect evaluation This evaluation category focuses on assessing generated samples from various specific aspects. Each task within this category typically focuses on a single type of evaluation. For this evaluation, we selected three tasks: SummEval (Fabbri et al., 2021) for text summarization, FED (Mehri and Eskenazi, 2020) for dialog-level and turn-level dialogue evaluation, we denote them as FED-D and FED-T, respectively. These tasks require ranking multiple samples based on their evaluation scores and measure the correlation between these scores and human judgments using metrics like Spearman, Pearson, and Kendall.

[1]For instance, the articles in OpenAI-Sum aspect are sourced from Reddit posts, whereas the articles in SumEval are derived from CNN/DM news sources.

Overall aspect evaluation This evaluation category involves assessing the overall quality of generated samples for arbitrary instructions. We chose MT-benchmark (Zheng et al., 2023) for validation, which includes up to 7 types of instructions and model-generated samples, covering areas such as writing, mathematical reasoning, and code generation. In this benchmark, pairs of samples with human expert preferences are provided, and the evaluation model is tasked with determining which sample is better from the aspect of overall helpfulness. Accuracy is used as the metric to judge the agreement between model judgments and the majority of human expert votes. MT-benchmark includes two stages of evaluation: Stage-1, which consists of a one-turn dialogue, and Stage-2, which involves a second-turn dialogue for evaluation.

Baselines For the detailed introduction to the baseline methods, please refer to Appendix 7.4

5 Results

5.1 Multi-aspect Evaluation

Overall Performance From results from Table 2, 3, and 4, it is clear that InstructEval, with both 7B and 13B parameters, exhibits competitive performance and often surpasses other commercial LLMs (GPTs) across a range of datasets on average. In the case of SummEval, the best-performing InstructEval surpasses ChaGPT-based G-EVAL by 14% and 12% in terms of Spearman and Kendall correlations, respectively. It also performs on par with GPT-4 based G-EVAL-4. On FED-D and FED-T, the best InstructEval outperforms the highest-performing GPTScore by 2.7% and 2.8%, respectively. Increasing the model's parameter size from 7B to 13B

Model	Coherence		Consistency		Relevance		Fluency		AVG	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
ROUGE-1	0.167	0.126	0.160	0.130	0.115	0.094	0.326	0.252	0.192	0.150
ROUGE-2	0.184	0.139	0.187	0.155	0.159	0.128	0.290	0.219	0.205	0.161
ROUGE-L	0.128	0.099	0.115	0.092	0.105	0.084	0.311	0.237	0.165	0.128
UniEval	0.575	0.442	0.446	0.371	0.449	0.371	0.426	0.325	0.474	0.377
BARTScore	0.448	0.342	0.382	0.315	0.356	0.292	0.356	0.273	0.385	0.305
G-EVAL†	0.440	0.335	0.386	0.318	0.385	0.293	0.385	0.293	0.399	0.310
G-EVAL-4†	0.582	0.457	0.507	0.425	0.547	0.433	**0.455**	**0.378**	0.523	0.423
GPTScore(davinci-003)	0.434	–	0.449	–	0.403	–	0.381	–	0.417	–
GPTScore (Llama 7B)	0.287	0.222	0.464	0.385	0.269	0.209	0.361	0.295	0.320	0.277
GPTScore (Llama-Chat, 7B)	0.303	0.234	0.473	0.392	0.298	0.222	0.371	0.303	0.361	0.288
Llama-Chat-ELS (7B)	0.182	0.137	0.261	0.215	0.138	0.105	0.301	0.254	0.220	0.178
InstructEval-ELS(7B)	0.639	0.507	0.527	0.441	0.427	0.326	0.302	0.247	0.474	0.380
w/o aspect	0.487	0.380	0.420	0.349	0.579	0.465	0.335	0.276	0.455	0.368
InstructEval-reg(7B)	0.572	0.452	0.506	0.422	0.519	0.412	0.386	0.318	0.496	0.401
w/o aspect	0.492	0.382	0.440	0.366	0.543	0.430	0.352	0.289	0.457	0.367
InstructEval-reg(13B)	**0.626**	**0.498**	**0.531**	**0.444**	0.557	0.442	0.384	0.319	**0.525**	**0.426**
w/o aspect	0.505	0.397	0.445	0.369	**0.577**	**0.457**	0.360	0.298	0.472	0.380

Table 2: Sample-level Spearman (ρ) Kendall (τ) correlations with human ratings on SummEval. Best results in each column are denoted **in Bold**. "-reg" and "-ELS" indicate using regression score and expected Likert score separately.† denotes results reported in the original paper. We highlight the columns of zero-shot aspects in gray.

yields a significant improvement in average performance. Specifically, there is a 1.9% and 2.5% increase in Spearman and Kendall on the SummEval, and a 2.5% and 1.7% increase in Spearman on the FED-D and FED-T respectively.

Regression Score vs Expected Likert Score Despite the continued training with the pretrained language model head, the expected Likert score (InstructEval-ELS) consistently exhibits inferior performance compared to the regression score (InstructEval-reg). We believe the reason behind this is that the limitations imposed by restricting the output spaces outweigh the benefits gained from reusing the language model head.

Finetuning with Overall vs Multi-aspects It is interesting to observe that *w/o aspects* (InstructEvals only trained with HPM data) are able to generalize to specific aspect definitions. Additionally, we discovered *w/o aspects* perform well on some aspects that correlates with informativeness. For instance, the performance of InstructEval-reg (13B, w/o aspects) on the relevance of the summary surpasses that of GPT-4. However, it is important to note that the improvements achieved in multi-aspect evaluation are limited. By incorporating annotated multi-aspect data, InstructEval improves by 5.3%, 1.6%, 2.6% in terms of Spearman on SummEval, FED-D and FED-T, respectively.

Zero-shot Ability on Aspects and Tasks In our research, while certain tasks and aspects, such as coherence and relevance in summarization, are directly annotated in the training set (e.g., OpenAI-sum aspect), most task and aspect combinations in the benchmarks are predicted in a near zero-shot manner. Specifically, for the finetuned aspects in SummEval, we noted average improvements of 4.1% and 6.2% in Spearman scores for the 7B and 13B versions of InstructEval-reg, respectively. It's important to note that while training samples from Helpful-HH also involve dialogue, their task formats differ from our evaluation methods in FED-D and FED-T. As indicated in Table 3 and 4, the annotated aspects show average improvements of 3.0% and 3.1% for the 13B model, respectively. Improvements for zero-shot aspects were observed at 0.1% and 2.2%, respectively. Overall, our findings suggest that multi-aspect eval-instruct not only significantly enhances directly annotated aspects but also improves zero-shot capabilities for unseen aspect definitions.

Model	COH	ERR	CON	DIV	DEP	LIK	UND	FLE	INF	INQ	AVG
GPTScore(davinci-001)†	0.569	0.457	0.329	**0.628**	**0.669**	**0.634**	0.524	0.515	**0.602**	0.503	0.543
GPTScore(davinci-003)†	0.134	0.094	0.181	-0.066	0.341	0.184	0.196	0.072	0.317	-0.101	0.135
GPTScore (Llama, 7B)	0.107	0.01	0.166	-0.215	-0.189	-0.051	0.130	-0.076	-0.084	0.077	-0.132
GPTScore (Llama-Chat, 7B)	0.146	0.035	0.152	-0.173	-0.112	0.035	0.187	-0.025	-0.005	0.157	0.031
Llama-Chat-ELS (7B)	0.301	0.117	0.114	0.161	0.210	0.121	0.198	0.257	0.223	0.200	0.190
InstructEval-ELS (7B)	0.613	0.485	0.419	0.439	0.490	0.579	0.504	0.607	0.504	0.465	0.511
InstructEval-reg (7B)	**0.683**	**0.530**	0.489	0.415	0.503	0.589	**0.628**	0.622	0.520	0.491	0.547
w/o aspect	0.649	0.514	0.465	0.446	0.485	0.585	0.577	0.611	0.480	0.475	0.529
InstructEval-reg (13B)	0.660	0.462	**0.497**	0.517	0.572	0.620	0.599	**0.648**	0.584	0.541	**0.570**
w/o aspect	0.643	0.501	0.470	0.472	0.552	**0.634**	0.563	0.618	0.535	**0.555**	0.554

Table 3: Spearman correlations with human ratings on dialog-level FED (FED-D). Best results in each column are denoted **in Bold**. We highlight the column of zero-shot aspects in gray.

Model	INT	ENG	SPE	REL	COR	SEM	UND	FLU	AVG
GPTScore(davinci-001)†	0.501	**0.496**	0.214	0.452	0.434	0.444	0.365	0.160	0.383
GPTScore(davinci-003)†	0.224	0.355	0.151	0.380	0.428	0.405	0.311	**0.367**	0.328
GPTScore (Llama 7B)	0.141	0.153	0.131	0.29	0.237	0.253	0.239	0.311	0.219
GPTScore (Llama-chat 7B)	0.132	0.151	0.079	0.276	0.227	0.251	0.233	0.257	0.201
Llama-Chat-ELS (7B)	0.266	0.157	0.180	0.097	0.091	0.125	0.057	0.123	0.137
InstructEval-ELS(7B)	0.325	0.329	**0.326**	0.474	0.449	0.405	0.399	0.202	0.363
InstructEval-reg(7B)	0.323	0.365	0.259	0.477	0.536	0.428	0.374	0.215	0.372
w/o aspect	0.339	0.388	0.268	0.451	0.512	**0.451**	0.370	0.167	0.368
InstructEval-reg(13B)	**0.421**	0.430	0.296	**0.503**	**0.557**	0.443	**0.399**	0.237	**0.411**
w/o aspect	0.384	0.427	0.317	0.477	0.521	0.381	0.379	0.193	0.385

Table 4: Spearman correlations with human ratings on turn-level FED (FED-T). Best results in each column are denoted **in Bold**. We highlight the column of zero-shot aspects in gray.

5.2 Overall Aspect Evaluation

The main focus of this evaluation is the generalization ability of diverse instructions, and we select the strong ChatGPT as our primary baseline. To utilize ChatGPT as an evaluator, we implement four different approaches: 1. ChatGPT-point: This approach applies a point-wise scoring method similar to InstructEval. We ask ChatGPT to predict a Likert score between 1 and 10. 2. Chat-pair: This approach employs a pairwise comparison method. We present ChatGPT with two options (sample A and sample B) and ask it to choose the better option. 3. ChatGPT-reverse: In this approach, we reverse the order of the options presented in Chat-pair. ChatGPT is asked to predict the preferred option between sample B and sample A. 4. ChatGPT-jointly combines both ChatGPT-pair and ChatGPT-reverse to eliminate position bias, similar to the approach used by Zheng et al. (2023).

Based on the results presented in Table 5, it is evident that InstructEval consistently demonstrates its superiority over ChatGPT-point in all settings using the same point-wise scoring approach. When comparing with the pairwise methods, InstructEval initially underperforms compared to ChatGPTs in Stage-1 but surpasses them in Stage-2. Increasing the parameter size only improves the performance in Stage-2 while being detrimental to Stage-1. We speculate that this is because the main training dialogue samples, Helpful-HH and Harmful-HH, primarily consist of multi-turn dialogues. Increasing the parameter size causes the model to focus more on capturing multi-turn features while potentially neglecting single-turn interactions. After adding multi-aspect eval-instruct, we observe that they are beneficial to the Stage 1 of overall aspects. Additionally, by only using the best contributing Helpful-HH dataset (according to Table 6), InstructEval-HH gains an further improvements in Stage-1. This highlights the untapped potential

Model	Stage-1		Stage-2	
	tie	w/o tie	tie	w/o tie
ChatGPT-point	0.47	0.51	0.50	0.51
ChatGPT-pair	0.59	0.70	0.58	0.72
ChatGPT-pair(reverse)	**0.59**	0.71	0.59	0.72
ChatGPT-pair(joint)	0.55	0.59	0.56	0.61
GPTScore(Llama 7B)	0.40	0.50	0.33	0.41
GPTScore(Llama-chat 7B)	0.38	0.48	0.38	0.47
Llama-Chat-ELS (7B)	0.46	0.58	0.44	0.55
InstructEval (7B)	0.50	0.64	0.56	0.70
w/o aspect	0.51	0.65	0.55	0.69
InstructEval (13B)	0.44	0.52	0.59	0.74
w/o aspect	0.48	0.60	**0.59**	**0.74**
InstructEval-HH (7B)	0.58	**0.73**	0.57	0.71

Table 5: Agreement with human expert on MT-benchmark. **tie** and **w/o tie** indicate whether using tie labels. We apply regression score as our default settings.

Model	Sum	FED-T	FED-D	MT	AVE
SHP	-0.210	0.179	0.290	0.345	0.151
Helpful-HH	0.310	0.354	**0.556**	**0.720**	0.485
Harmful-HH	-0.126	0.142	-0.230	0.311	0.024
Instruct-GPTJ	0.479	**0.541**	0.318	0.375	0.428
OpenAI-Sum	0.461	0.533	0.322	0.645	0.490
WebGPT	0.060	0.148	0.317	0.705	0.308
All	0.457	0.368	0.529	0.630	0.496
All+Sum-asp	**0.514**	0.328	0.516	0.685	0.511
All+ALL-asp	0.496	0.372	0.547	0.670	**0.521**

Table 6: Spearman of InstructEval independently fine-tuned on various datasets.

in optimizing the data distribution of HPM, which we will leave for future work.

5.3 Ablation Study on Training Data

To gain a comprehensive understanding of how different types of data impact the performance, we conducted a comprehensive ablation study on the composition of the training set. The first block of Table 6 presents the independent performance of training Llama using each HPM dataset individually. In the second block, "ALL" represents the utilization of a mixture of all the datasets in the first block. "+Sum-asp" further incorporate "OpenAI-sum asp" alongside the "ALL" dataset, and "+ALL-asp" adds both "OpenAI-sum asp" and our constructed "Multi-aspect eval-instruct". The results of SumEval(Sum), FED-T, and FED-D are average Spearman of different aspects. MT reports the average performance of stage-1 and stage-2 without "tie" labels.

Ablation on HPM Data Results in Table 6 reveals significant variations across datasets in performance, despite all samples being labeled under the general aspect of "overall helpfulness." Merely having a large amount of data, "SHP" for example, does not guarantee superior performance. Surprisingly, OpenAI-Sums, trained exclusively on summarization data, exhibits strong performance across multiple tasks, including summarization, turn-level dialogue, and overall aspect evaluation. However, its performance in dialog-level evaluation is comparatively weaker due to the absence of multi-turn dialogue in training. On the other hand, training with "All" datasets achieves a well-balanced performance across all evaluation criteria, indicating its effectiveness in handling diverse tasks. In real-world scenarios, where testing samples often exhibit greater diversity than the benchmarks, training with a comprehensive datasets encompassing all relevant aspects is likely to lead to improved performance and significance.

Multi-Aspect Data The incorporation of specific aspect datasets consistently enhances performance in both overall and specific aspects evaluation (second block vs first block in Table 6). Notably, the inclusion of constructed "Multi-aspect instruct" data demonstrates a stable improvement in all multi-aspect evaluation ("+All-asp" vs "+Sum-asp").

6 Conclusion and Future Research

This paper has addressed the challenges of text evaluation with LLMs. The major insight of this study lies in leveraging HPM resources for text evaluation. By utilizing the abundant HPM datasets, the study demonstrates the potential of fine-tuning LLMs to evaluate specific aspects and overall quality in a zero-shot manner. Additionally, this paper propose to use a multi-aspect instruct set constructed by GPT-4, to enhance the performance of multi-aspect evaluation.

The insights gained from the analysis of resource contributions in this paper can guide future research in designing more effective evaluation methodologies. Understanding the specific benefits provided by different resources can inform the selection and utilization of appropriate datasets for evaluation tasks.

Limitations

The evaluation of LLMs heavily relies on the availability of labeled samples for fine-tuning. In this study, we focus on leveraging human preference modeling (HPM) resources for evaluation. However, the HPM datasets used may not cover the full range of evaluation aspects required for comprehensive text evaluation. The reliance on specific HPM datasets limits the generalizability of the proposed approach to evaluate a broader range of tasks and aspects. Future research should explore strategies to incorporate more diverse and representative labeled samples to enhance the evaluation capabilities.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Kawin Ethayarajh and Dan Jurafsky. 2021. Utility is in the eye of the user: A critique of nlp leaderboards.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

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

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9332–9346. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. *arXiv preprint arXiv:2006.12719*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Sujian Li, and Yajuan Lyu. 2023. Wecheck: Strong factual consistency checker via weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 307–321. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

7 Appendix

7.1 Multi-aspect Data Construction

Table 8 lists all the aspects and their definitions along with their appeared datasets. These aspect definitions were initially generated by ChatGPT, as described in Table 7, and subsequently revised and curated by human experts.

7.2 Training Data Details

The overall statistics of preprocessed training datasets are listed in Table 1. The open-sourced HPM datasets are Stanford Human Preference (SHP)[2], WebGPT[3], OpenAI-Sum[4], Helpful-HH and Harmful-HH[5], Instruct-GPTJ [6], and an additional Openai-Sum aspect. More details of our training data is provided in Appendix 7.2.

For SHP, which additionally provide confidence annotations, we apply a filtering criterion where we exclude samples with confidence ratios smaller than 5 to ensure data quality. For our constructed Multi-aspect instruction samples, the instructions are randomly selected from SHP, WebGPT and Helpuful-HH and. For OpenAI-sum, since it provide additional labeled specific aspect preferences as a validation split, we directly incorporate it in training set. Overall, the training set contains 727,070 eval-instruct, 655,101 of them are overall aspect, only 3,500 of them are specific aspects annotated by GPT-4.

7.3 Model Training Details

We fixed the hyper-parameters for all models with a learning rate of $1e-5$ with no weight decay and linear warmup for 3% of the total training steps. We use a batch size of 256, and a maximum sequence length of 1024 and truncate exceeded tokens. The overall dataset is trained for only one epoch on 8 NVIDIA A800 gpus with 80GB of VRAM.

7.4 Baseline Methods

Our primary baselines consist of evaluation models built on large language models (LLMs), with a particular focus on GPTs. For instance, GPT-score (Fu et al., 2023) is used as a baseline for text generation evaluations, while G-EVAL (Liu et al., 2023) serves as a baseline for text summarization. Also, we apply our base model, Llama, as baseline.

[2] https://huggingface.co/datasets/stanfordnlp/SHP

[3] https://huggingface.co/datasets/openai/webgpt_comparisons

[4] https://huggingface.co/datasets/openai/summarize_from_feedback

[5] https://huggingface.co/datasets/Anthropic/hh-rlhf

[6] https://huggingface.co/datasets/Dahoas/synthetic-instruct-gptj-pairwise

My Prompt: list some aspect when we need to evaluate the response to a post is good enough.
ChatGPT Response: Evaluating the quality of a response to a post can be subjective, but here are some generally accepted aspects to consider: **Relevance**: The response should directly address the topic or question raised in the original post. Irrelevant responses can distract from the conversation and are generally not considered high-quality. **Accuracy**: If the response includes factual information, it should be correct. Misinformation can mislead readers and degrade the quality of the conversation. **Completeness**: A good response should fully address the question or topic at hand. It's not necessary to write a book, but the response should be detailed enough to provide a satisfactory answer or contribution. **Clarity:** The response should be easy to understand. This involves good grammar, clear sentence structure, and the appropriate use of terminology. **Respectfulness**: Even in disagreements, a quality response will respect other participants in the conversation. This means avoiding personal attacks, offensive language, and other types of disrespectful behavior. **Substantiated Arguments**: If the response is presenting an opinion or argument, it should be backed by relevant evidence or reasoning. Unsupported claims can reduce the quality of a response. **Originality**: Responses that bring new ideas, perspectives, or information to the conversation are often considered higher quality. This doesn't mean every response needs to be groundbreaking, but merely parroting what others have already said may not add much value. **Engagement**: The response should invite further conversation. This could be through asking follow-up questions, suggesting new lines of inquiry, or otherwise indicating a willingness to continue the dialogue. **Timeliness**: A response is more valuable if it's given when the conversation is still active. While some topics are evergreen, others may lose relevance over time. **Conciseness**: While completeness is important, so is brevity. A good response communicates its points efficiently without unnecessary filler.

Table 7: A example of using ChatGPT to list out evaluation asepcts and their definitions.

Because Llama without instruction tuning cannot follow instructions to evaluate, we adopt the scoring approach of GPT-score, i.e. scoring the target output by its perplexity. We also test Llama-Chat, the version of Llama trained with RLHF, in both GPTScore and ELS way of scoring.

Aspect	Datasets	Definition
Relevance	SHP, Helpful-HH, Instruct-GPTJ, WebGPT	The output should directly address the topic or question raised in the instruction or input. Irrelevant output can distract from the conversation and are generally not considered high-quality.
Relevance	OpenAI-Sum asp	Consider whether all and only the important aspects are contained in the summary.
Accuracy	SHP,Helpful-HH, Instruct-GPTJ, WebGPT	If the output includes factual information, it should be correct. Misinformation can mislead readers and degrade the quality of the conversation.
Consitentcy	OpenAI-Sum asp	Consider whether the summary does reproduce all facts accurately and does not make up untrue information.
Coherence	Helpful-HH, OpenAI-SUM asp	Does the answer demonstrate logical and smooth progression of ideas? Are the statements and arguments connected in a cohesive and meaningful way?
Completeness	SHP,Helpful-HH, Instruct-GPTJ, WebGPT	A good output should fully address the question or topic at hand. It's not necessary to write a book, but the output should be detailed enough to provide a satisfactory answer or contribution.
Interesting	Helpful-HH	The ouput should be interesting enough for the reader to read.
Depth	Helpful-HH, Instruct-GPTJ, WebGPT	Does the answer offer a thoughtful and insightful analysis of the question or topic? Does it go beyond superficial or obvious information to provide deeper understanding or valuable insights?
Clarity	Instruct-GPTJ, WebGPT	Is the answer clear, understandable, and well-organized? Is it presented in a coherent manner that is easy to follow?
Fluency	Helpful-HH, Instruct-GPTJ, WebGPT	Fluency measures the quality of individual sentences, are they well-written and grammatically correct.
Informative	Helpful-HH	Is the system informative throughout the conversation?
Understandable	Helpful-HH	The response should be easy to understand. This involves good grammar, clear sentence structure, and the appropriate use of terminology

Table 8: Aspects and their definitions used for constructing multi-aspect instruction data.