Automatic Evaluation of Generative Models with Instruction Tuning

Shuhaib Mehri[1] **and Vered Shwartz**[1,2]
[1] University of British Columbia
[2] Vector Institute for AI
shuhaibm@student.ubc.ca, vshwartz@cs.ubc.ca

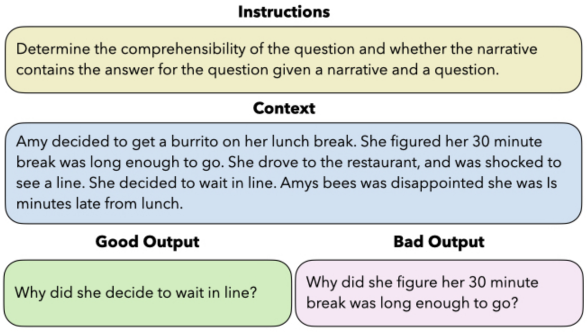
Abstract

Automatic evaluation of natural language generation has long been an elusive goal in NLP. A recent paradigm fine-tunes pre-trained language models to emulate human judgements for a particular task and evaluation criterion. Inspired by the generalization ability of instruction-tuned models, we propose a learned metric based on instruction tuning. To test our approach, we collected HEAP, a dataset of human judgements across various NLG tasks and evaluation criteria. Our findings demonstrate that instruction tuning language models on HEAP yields good performance on many evaluation tasks, though some criteria are less trivial to learn than others. Further, jointly training on multiple tasks can yield additional performance improvements, which can be beneficial for future tasks with little to no human annotated data.

1 Introduction

Natural language generation (NLG) has made significant leaps forward in recent years thanks to large language models (LLMs; Brown et al., 2020; Open, 2023). Yet, to date, there is no standard evaluation protocol for NLG systems. Human evaluation provides the most accurate assessment, but its costly and time-consuming nature makes it less practical for large-scale evaluations, and it's rarely conducted as part of the system development cycle. For this reason, automatic evaluation metrics have been widely adopted. The majority of automatic metrics compare the system outputs against a set of reference texts, measuring either lexical overlap (e.g., Papineni et al., 2002; Lin, 2004) or semantic similarity (e.g., Zhang et al., 2019).

Reference-based metrics suffer from many drawbacks. First, system outputs that are different from the references are scored low, even if they are correct. Second, multiple studies have noted poor correlation with human judgements (Novikova et al., 2017; Dhingra et al., 2019; Chen et al., 2019; Kryscinski et al., 2019). Third, methods that were designed with one task in mind, such as BLEU (Papineni et al., 2002) for machine translation and ROUGE for summarization (Lin, 2004), don't necessarily transfer well to other tasks (Liu et al., 2016; Nema and Khapra, 2018). Finally, by producing a single score based on similarity to the references, some important but more nuanced dimensions might be missed, such as faithfulness, answerability, and more.



Instructions

Determine the comprehensibility of the question and whether the narrative contains the answer for the question given a narrative and a question.

Context

Amy decided to get a burrito on her lunch break. She figured her 30 minute break was long enough to go. She drove to the restaurant, and was shocked to see a line. She decided to wait in line. Amys bees was disappointed she was Is minutes late from lunch.

Good Output

Why did she decide to wait in line?

Bad Output

Why did she figure her 30 minute break was long enough to go?

Figure 1: Example from HEAP, originally taken the TellMeWhy dataset (Lal et al., 2021), here focusing on the question answerability (QA) criteria.

A recent alternative approach is learned metrics. Such metrics leverage a pre-trained language model and fine-tune it to emulate human judgements (e.g.; Sellam et al., 2020; Zhao et al., 2020). Learned metrics are typically tailored to specific tasks (e.g., machine translation) and criteria (e.g., similarity to the references), and they can be reference-based or reference-less.

In this work, we propose to train reference-less learned metrics using instruction tuning. Instruction tuning involves presenting the model with natural language instructions in addition to the task inputs. Including the instructions as part of the input enables models to generalize better, perform well in zero-shot and few-shot settings (Wei et al., 2021; Gupta et al., 2022), and better align with human values (Peng et al., 2023).

Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 42–52
December 6, 2023 ©2023 Association for Computational Linguistics

To train our metric, we collected the **H**uman **E**valuations of **A**nswer **P**airs dataset (HEAP). HEAP was composed from the human evaluation results for 8 datasets, along 22 diverse evaluation criteria, such as comprehensibility, appropriateness, grammaticality, and informativeness, as detailed in Table 1.[1] We converted all data points to a uniform comparative format, consisting of the task instructions, and two context-generation pairs, such that one generation (good in Figure 1) was ranked better than the other (bad).

We used HEAP with instruction tuning in single-task, multi-task, and cross-task setups. We find that most criteria are learnable, though more nuanced or complex ones (e.g., answer validity) are more difficult to learn than others (e.g., grammaticality). We also show that fine-tuning on the task is essential, and that multi-tasking can help with the more difficult tasks. Finally, the cross-task setup is less successful, but can be improved by training only on a subset of similar tasks to the target task.

We hope that our findings will guide future research on automatic evaluation for NLG systems.[2]

2 Related Work

Automatic Evaluation of Generative Tasks. Numerous automatic methods exist for evaluating generative models. The majority of metrics involve assessing the similarity between a generated output and a reference text. Commonly used metrics include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), which focus on measuring lexical overlap between generated outputs and a reference. More recent methods, such as BERTScore (Zhang et al., 2019), go beyond lexical overlap by embedding both the generations and the references into a shared space and computing cosine similarity between the embeddings. All these metrics operate at the surface level, predominantly focusing on lexical similarity.

Some metrics have been proposed which are trained to emulate human judgements. BLEURT (Sellam et al., 2020) is a BERT-based metric which is first trained to estimate the scores from existing automatic metrics for a large number of synthetic sentence pairs, and then trained to emulate human judgements for a machine translation task. Similarly, Zhao et al. (2020) proposed a RoBERTa-based metric for dialogue evaluation which is first trained on a large number of sentence pairs with a next sentence prediction objective, and then trained on a small number of human annotations for the task. Learned metrics are tailored to specific tasks. They can take on different forms: reference-based, where the metric is trained to compare the system's output to a reference text, like in BLUERT; reference-less, where the metric scores the output along some criterion without the use of references (Sinha et al., 2020); or a combination of both, as seen in Ghazarian et al. (2019)'s work. In this work, we propose a reference-less learned metric and investigate the transferability between different tasks and criteria.

Instruction Tuning. Instruction tuning is a fine-tuning technique that involves training a model on a variety of tasks, leveraging natural language instructions to guide the model towards producing the correct answers. Recent studies have showcased the effectiveness of this technique in improving LLMs' ability to generalize in a zero-shot and few-shot setting (Chen et al., 2022; Wei et al., 2021; Peng et al., 2023). Most pertinent to our work, Gupta et al. (2022) applied instruction tuning to 48 dialogue-related tasks, including dialogue evaluation. They showed that their instruction-tuned dialogue evaluation metric achieves improved correlation with human judgements, even in a cross-task setup when training on other dialogue tasks. In this work, we use instruction-tuning to train automatic evaluation metrics for a diverse set of tasks and criteria. The use of instructions allows for more transferability between different tasks and criteria, and could be beneficial when data for a particular task is sparse.

3 Dataset

We introduce the **H**uman **E**valuations of **A**nswer **P**airs (HEAP) dataset. HEAP is designed to train and evaluate automatic methods for the evaluation of generative tasks. It is derived from existing human evaluations that were performed on 8 generative tasks detailed in Table 1. We obtained the data from public releases as well as by reaching out to the authors of the respective papers. Appendix B provides examples for each task and criteria along with the instructions we used for them.

The original human evaluations for some of the datasets included comparative evaluation (i.e.,

[1] The annotations were generously shared with us by the dataset creators.

[2] Code and data available at https://github.com/Shuhaibm/heap

Task	#Examples	Dataset
Advice Helpfulness [AH]	1,200	TuringAdvice: Advice Generation (Zellers et al., 2021)
Answer Grammaticality [AG]	598	TellMeWhy: Answering Why-Questions (Lal et al., 2021)
Answer Validity [AV]	598	
Question Answerability [QA]	1,917	
HellaSwag [HS]	149,841	HellaSwag: Commonsense NLI (Zellers et al., 2019)
Commonsense Reasoning [CR]	1,079	CommonGen: Commonsense Reasoning (Lin et al., 2020)
Best Counter Narrative [BCN]	1,000	Counter Narratives Against Hate Speech (Tekiroğlu et al., 2022)
Choose-or-not [CCN]	884	
Grammaticality [CNG]	863	
Specificity [CNSp]	1,139	
Suitability [CNSu]	1,471	
Counter Narrative Informativeness [CNI]	783	CHASM: Countering Online Hate Speech and Microaggressions (Ashida and Komachi, 2022)
Counter Narrative Offensiveness [CNO]	685	
Counter Narrative Stance [CNSt]	724	
Hate Speech Offensiveness [HSO]	29,970	
Story Rewriting Counterfactual [SRC]	4,400	TimeTravel: Counterfactual Story Rewriting (Qin et al., 2019)
Story Rewriting Ending [SRE]	4,400	
Story Rewriting Plot [SRPl]	4,400	
Story Rewriting Premise [SRPr]	4,400	
Story Rewriting Second [SRS]	4,400	
Attenuator Effectiveness [DIA]	7,176	Defeasible Inference (Rudinger et al., 2020)
Intensifier Effectiveness [DII]	7,176	

Table 1: Human evaluation criteria (referred to as "tasks" in this paper) included in HEAP.

which of the answers is better along some criterion), while others included absolute scores of an answer's quality. We decided to go with the comparative setup based on the findings of Askell et al. (2021) and Bai et al. (2022) who demonstrated that a ranked preference model, which is a model trained to assign a higher score to the 'better' sample in a given pair, outperforms other training objectives like imitation learning and binary discrimination. To that end, we converted absolute scores and comparison between multiple answers into pairwise comparisons.

The dataset contains 229,104 instances. The instances from each task are randomly split into 80% train, 10% validation, and 10% test sets and combined. Each data point in HEAP consists of two generated outputs, `good_sample` and `bad_sample`, where each sample has its own context `C`. Each data point belongs to a "task", which is a combination of the original dataset (e.g. advice generation) and evaluation criterion (e.g. advice helpfulness). An example data point can be seen in Figure 1.

4 Method

We propose to fine-tune pre-trained language models to predict a scalar score for text outputs along various criteria. We train the models using natural language instructions (Sec 4.1) and investigate the extent that this setup allows for out-of-domain generalization for new tasks (Sec 4.2).

4.1 Instruction Tuning

Instruction tuning refers to a setup in which natural language instructions are prepended to the input (Figure 1). By incorporating instructions in a model's training, it learns how to arrive at the expected output for a given task (Mishra et al., 2022).

To find the optimal instructions for each task, we manually wrote a diverse set of instructions and chose the instruction that yielded the best performance on the task's validation set. Details about the instructions used can be found in Appendix B.

We used BART-base (Lewis et al., 2020), a pre-trained encoder-decoder model with 140M parameters, for all our experiments. We fine-tuned BART to predict a score for each answer. Specifically, the input for each example is in the following format: `<instructions> <context> <answer>`. We embed the input using BART and feed the last hidden state into a linear layer to predict a scalar score r, where a higher score denotes a more favourable input. Following prior work (Christiano et al., 2017; Askell et al., 2021), we maximize the difference between the scores of the good and bad outputs with the following loss function:

$\mathcal{L} = \log(1 + \exp(r_{\text{bad}} - r_{\text{good}}))$.

4.2 Evaluation Setups

We train and evaluate the models in the following setups:

Single-Task. In this setup, for a target task t, we train the model on the training set composed of only t's instances (D^t_{train}) and test it on the test set composed of only t's instances (D^t_{test}).

Multi-Task. In this setup, we investigate whether the different tasks can mutually benefit each other. We train a single model on the entire HEAP training set (D_{train}) and test it on the test set composed of only t's instances (D^t_{test}).

Cross-Task. In this setup, we investigate our instruction-tuned models' zero-shot generalization abilities, by evaluating them on unseen tasks. For a target task t, we train a model on the HEAP training set excluding t's instances ($D^{/t}_{\text{train}}$), and test it on the test set composed of only t's instances (D^t_{test}). We hypothesized that the model would be able to generalize to a new task by learning to follow instructions.

Cross-Cluster. In this setup, we repeat the cross-task setup, but train the model on a subset of HEAP. We refer to each such subset as a "cluster". Each cluster consists of handpicked tasks based on certain similarities. For a target task t that belongs to cluster C, we train a model on the cluster's training set excluding t's instances ($C^{/t}_{\text{train}}$), and test it on the test set composed of only t's instances (C^t_{test}). We hypothesize that being more selective with tasks will further improve a model's ability to generalize to a new task.

5 Experimental Setup

Baselines. Other than the single-task, multi-task, cross-task, and cross-cluster setups described in Sec 4.2, we also included the **base** setup, in which we used BART off-the-shelf without fine-tuning it.

Hyper-parameter Tuning. We performed hyper-parameter tuning on the validation set to select values for the following: learning rate ($2e-5$, $2e-4$, $3e-4$), gradient accumulation (4, 8, 16, 32, 64, 128), number of epochs ($1-20$), truncation, instructions, and labelling elements of the input. The selected values are available in Appendix A.

Evaluation Metrics. We evaluated the performance of our models using two metrics. The first metric is accuracy on the respective test set. That is, we obtained scores r_{good} and r_{bad} for the respective answers, and counted the percent of instances for which r_{good} was greater than r_{bad}. The second metric is Spearman rank-order correlation between the scores outputted by the model and the original human evaluation scores. This metric shows the extent to which the model's preferences align with human preferences.

6 Results

Table 2 presents the main results. We observe the following.

Fine-tuning is essential. The base model has an average accuracy of 50.58% across tasks, which is akin to a random baseline. The single task setup substantially improves upon the base model with an average of 63.24%.

Most criteria are learnable, as evident by the 12.66% difference in accuracy between the base and the single-task models. However, for a few tasks, even the best performance remains relatively low: SRC, SRE, SRPr, and SRS. These tasks all come from the TimeTravel dataset of counterfactual story rewriting (Qin et al., 2019) and they are inherently difficult, as they require comparing two almost identical stories along various dimensions.

Multi-tasking is beneficial. On average, the multi-task setup achieves 68.82% accuracy, 5.58% higher than the single-task setup. This indicates that there is transfer learning among the different tasks. Perhaps trivially, multi-tasking is especially beneficial when the single-task accuracy is low. The performance of tasks such as CNSt, CNG, CR, and CNI that already achieve good performance in the single-task setup either decreases or increases very slightly. Conversely, multi-tasking is the most beneficial for tasks that achieve low single-task performance, such as AV and BCN.

Success in the cross-task setup varies. The cross-task setup performs substantially worse than the multi-task setup (54.85% compared to 68.82% on average), which is expected since the target task training data is excluded. Compared to the single-task setup, the cross-task setup is beneficial for CCN, CNSu, SRE, BCN, and AV, but even in those cases, it is less beneficial than the multi-task setup.

Task	Accuracy				Spearman Rank-order Correlation			
	Base	Single-task	Multi-task	Cross-task	Base	Single-task	Multi-task	Cross-task
AH	52.55±8.02	**67.33±1.60**	66.94±1.27	55.84±7.22	-	-	-	-
AG	58.97±14.66	69.47±5.46	**78.07±6.62**	58.12±3.91	0.181	**0.379**	0.231	0.146
AV	53.9±2.66	44.35±10.03	**65.22±5.75**	58.87±10.92	0.063	-0.026	**0.293**	0.093
QA	46.35±14.11	58.96±14.58	**75.17±2.10**	42.19±3.76	-	-	-	-
HS	49.76±0.62	**67.91±0.42**	65.62±0.45	51.19±0.10	-	-	-	-
CR	49.07±5.29	77.96±4.37	**79.32±1.07**	54.32±3.86	-	-	-	-
BCN	47.33±2.05	50.60±3.29	**79.32±1.73**	63.00±2.00	-	-	-	-
CCN	44.19±4.62	64.54±12.48	**68.18±1.14**	68.16±1.72	-	-	-	-
CNG	56.32±2.48	77.67±2.24	**82.17±1.78**	62.84±2.89	0.089	0.472	**0.538**	0.193
CNSp	51.17±3.94	54.39±4.85	**64.03±3.16**	48.54±3.95	0.086	0.211	**0.278**	0.090
CNSu	43.24±5.06	58.64±6.23	**68.03±2.45**	64.64±1.03	-0.152	0.143	**0.413**	0.081
CNI	41.77±5.76	**83.59±2.11**	76.92±3.39	70.88±4.39	-0.081	**0.574**	0.472	-0.014
CNO	55.07±9.47	67.35±6.19	**69.12±1.47**	28.50±3.64	0.181	0.440	**0.580**	-0.125
CNSt	47.04±3.92	**76.39±2.78**	71.30±5.61	51.60±4.18	-0.087	0.436	**0.461**	0.129
HSO	46.15±7.72	**68.81±2.94**	66.43±2.53	49.55±0.76	-0.170	**0.425**	0.399	0.086
SRC	44.81±3.37	51.80±2.74	**57.38±1.64**	48.09±6.21	-	-	-	-
SRE	37.5±6.36	49.64±9.81	**58.93±5.36**	56.55±5.16	-	-	-	-
SRPl	45.89±20.37	70.72±1.21	**75.36±6.31**	50.24±6.03	-	-	-	-
SRPr	40.67±5.73	49.60±7.40	**56.00±3.46**	44.67±3.06	-	-	-	-
SRS	52.22±2.83	55.33±7.40	**61.67±3.34**	55.00±5.00	-	-	-	-
DIA	48.24±2.61	65.43±6.25	**69.68±1.90**	62.03±2.29	-0.009	0.302	**0.321**	0.268
DII	49.93±2.43	60.82±3.65	59.26±0.71	**61.84±5.09**	0.060	**0.252**	0.215	0.080

Table 2: Accuracy on the test set, and Spearman rank-order correlation with human judgements, for each task in each of the setups detailed in Sec 4.2. Accuracy is reported as the average of 5 runs with different random seeds. Correlation is reported for datasets that have ranked data. Bold indicates best performance and underline indicates second-best. **Takeaways**: (i) fine-tuning is essential; (ii) training on additional tasks is beneficial for most target tasks; (iii) success in the cross-task setup varies a lot.

For CNO, the cross-setup performed substantially worse even than the baseline, but we couldn't find a reasonable explanation for this behavior.

The number of per-task examples is not the most important factor. Notably, the number of examples available for each task had very weak correlations with the single-task performance (Pearson $\rho = 0.15$), the gain from multi-tasking ($\rho = -0.25$), and the gain from the cross-task setup ($\rho = 0.16$). We conclude that among the important indicators for good performance are the ease of the task, i.e., "easy" tasks such as verifying grammaticality can already achieve good performance without training on additional tasks.

Choosing the right tasks for transfer matters. Results for the cross-cluster setup is presented in Table 3. The unsurprising finding is that one can benefit from training on a cluster that consists of similar tasks. For example, the first cluster consists of tasks that require deep semantic understanding of the context C. The tasks in this cluster are diverse, ranging from advice helpfulness through general commonsense reasoning to defeasible and counterfactual reasoning. As a result, the average accuracy for the tasks in this cluster drops from 55.26% to 50.03%.

Conversely, when the clusters involve more closely-related tasks, it is beneficial to limit the training to the cluster tasks. For example, tasks that require more superficial understanding of the context C or none at all, involve evaluating the grammaticality, specificity, suitability, informativeness, offensiveness, and stance of the generated answers. Those tasks are related enough to increase the average accuracy from 52.96% to 55.67%. When further focusing on tasks coming from similar datasets, such as tasks pertaining to hate speech detection, the performance improvement is more substantial (56.41% to 60.3%). A similar trend holds when focusing on different criteria from the same dataset, e.g. from 48.66% to 53.06% on answering why-questions, 60.56% to 61.94% on defeasible inference, and 50.91% to 53.22 on counterfactual story rewriting.

7 Conclusion

We proposed to use instruction tuning to learn automatic evaluation metrics. To test the effectiveness of this approach, we introduced HEAP, a collection of human judgements along diverse dimensions for various generative tasks. Our experiments confirm

Task	Single-Task	Cross All	Cross Cluster
Cluster 1: Require understanding the context C			
AH	67.33±1.60	55.84±7.22	**56.39±2.08**
AV	44.35±10.03	**58.87±10.92**	53.9±1.00
QA	58.96±14.58	42.19±3.76	**52.78±5.79**
HS	67.91±0.42	**51.19±0.10**	44.46±0.44
CR	77.96±4.37	**54.32±3.86**	48.77±1.90
BCN	50.60±3.29	**63.00±2.00**	58.00±2.16
SRC	51.80±2.74	**48.09±6.21**	41.53±4.70
DIA	65.43±6.25	**62.03±2.29**	43.86±2.57
DII	60.82±3.65	**61.84±5.09**	50.61±4.45
Average	-	55.26	50.03
Cluster 2: Don't require understanding the context C			
AG	69.47±5.46	58.12±3.91	**65.81±8.46**
CNG	77.67±2.24	**62.84±2.89**	56.32±1.63
CNSp	54.39±4.85	48.54±3.95	**51.17±1.49**
CNSu	58.64±6.23	**64.64±1.03**	54.96±3.04
CNI	83.59±2.11	**70.88±4.39**	54.85±5.31
CNO	67.35±6.19	28.50±3.64	**43.96±3.62**
CNSt	76.39±2.78	51.60±4.18	**65.75±3.87**
HSO	68.81±2.94	49.55±0.76	**58.06±3.48**
SRPl	70.72±1.21	**50.24±6.03**	47.83±1.18
SRPr	49.60±7.40	44.67±3.06	**58.00±5.89**
Average	-	52.96	55.67
Cluster 3: Hate speech related tasks			
BCN	50.60±3.29	63.00±2.00	**69.00±4.58**
CCN	64.54±12.48	**68.16±1.72**	64.05±4.05
CNG	77.67±2.24	62.84±2.89	**63.22±4.6**
CNSp	54.39±4.85	48.54±3.95	**59.36±0.51**
CNSu	58.64±6.23	**64.64±1.03**	**64.64±2.17**
CNI	83.59±2.11	**70.88±4.39**	57.00±2.47
CNO	67.35±6.19	28.50±3.64	**45.38±4.68**
CNSt	76.39±2.78	51.60±4.18	**67.58±2.85**
HSO	68.81±2.94	49.55±0.76	**52.50±1.48**
Average	-	56.41	60.30

Table 3: Per-task accuracy when the model is trained on all other tasks in the cross-task setup (**Cross All**) vs. all other tasks in the same cluster (**Cross Cluster**).

the importance of fine-tuning for developing metrics that align with human judgements. Further, we showed the advantage of fine-tuning on multiple tasks, and that a cross-task (zero-shot) setup yields positive results when trained on selected tasks. Collectively, our experiments reveal the value of instruction tuning in the domain of automatic evaluation of generative tasks. We hope that our findings will serve as a catalyst for inspiring future research on this topic.

Limitations

Task Balance. The number of examples in HEAP is imbalanced across tasks, as can be seen in Table 1. The number of examples range from 598 for AG and AV to 149,841 for HS. In preliminary experiments we tried to obtain a more balanced dataset by removing HS from cluster 1 (Table 3). This resulted in a drop of one point in average accuracy, but a significantly shorter training time. In the future, we will explore the possibility of obtaining more annotations for "lower-resource" tasks, applying data augmentation methods, or using more sophisticated multi-tasking techniques to overcome task imbalance.

Inherent Subjectivity. Our dataset is based on annotators' judgements of model-generated outputs along various dimensions. It's possible that some tasks involve inherent subjectivity, thus creating variance in the quality and consistency of the data for different tasks. This could further explain why our models were able to achieve better performance on more objective tasks, such as grammaticality judgement (Sec 6).

Ethics Statement

Data. The HEAP dataset is a compilation of human evaluations. We obtained them from public releases as well as by reaching out to the authors of the dataset papers. We plan to make it publicly available with the consent of the authors that contributed data. The annotations in the dataset do not include any personal information of the annotators. Details about the compensation for the annotators is available in the original papers. Finally, the contexts in HEAP come from diverse datasets (Table 1), some of which may include offensive, hateful, or sexual content. We did not perform quality control beyond what was performed by the original dataset creators.

Models. The HEAP dataset contains human judgements along various tasks, which may exhibit societal biases. Given that our evaluation models are trained to emulate these human judgements, it is possible that our models replicate these undesired biases.

Acknowledgements

We are sincerely thankful for the authors of the datasets used in this paper for sharing the human evaluation results with us. This work was funded, in part, by the Vector Institute for AI, Canada CIFAR AI Chairs program, an NSERC discovery grant, and a research gift from AI2.

References

Mana Ashida and Mamoru Komachi. 2022. Towards automatic generation of messages countering online hate speech and microaggressions. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 11–23, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Satanjeev Banerjee and Alon Lavie. 2005. 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*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Sarik Ghazarian, Johnny Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 82–89, Minneapolis, Minnesota. Association for Computational Linguistics.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

AI Open. 2023. Introducing chatgpt.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053, Hong Kong, China. Association for Computational Linguistics.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L. Hamilton, and Joelle Pineau. 2020. Learning an unreferenced metric for online dialogue evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2430–2441, Online. Association for Computational Linguistics.

Serra Sinem Tekiroğlu, Helena Bonaldi, Margherita Fanton, and Marco Guerini. 2022. Using pre-trained language models for producing counter narratives against hate speech: a comparative study. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3099–3114, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. 2021. TuringAdvice: A generative and dynamic evaluation of language use. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4856–4880, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. 2020. Designing precise and robust dialogue response evaluators. In *Proceedings of the 58th Annual Meeting of* *the Association for Computational Linguistics*, pages 26–33, Online. Association for Computational Linguistics.

A Hyper-Parameters

Task	**Truncate Right**	**Label Input**	**Gradient Accumulation**	**Learning Rate**	**#Epochs**
			Single-Task Setup		
AH	✓	✗	4	2e-5	17
AG	✓	✓	8	2e-5	15
AV	✓	✓	32	3e-4	10
QA	✓	✗	4	2e-4	20
HS	✓	✗	64	2e-5	17
CR	✓	✗	16	2e-4	18
BCN	✓	✓	4	3e-4	3
CCN	✓	✓	32	2e-4	15
CNG	✓	✓	32	2e-4	12
CNSp	✓	✗	8	2e-4	12
CNSu	✓	✓	128	3e-4	13
CNI	✓	✗	32	2e-4	15
CNO	✓	✓	64	3e-4	5
CNSt	✓	✗	4	2e-5	13
HSO	✓	✓	128	3e-4	2
SRC	✓	✓	64	2e-4	7
SRE	✓	✓	16	3e-4	2
SRPl	✓	✓	32	2e-4	15
SRPr	✓	✗	8	3e-4	4
SRS	✓	✗	8	2e-5	18
DIA	✓	✗	32	2e-4	15
DII	✓	✗	64	2e-5	17
			Multi-Task Setup		
Overall	-	-	8	2e-5	16
			Cross-Task Setup		
AH	✓	✗	4	2e-5	16
AG	✓	✓	8	2e-5	19
AV	✓	✓	16	3e-4	17
QA	✓	✗	32	2e-5	9
HS	✓	✗	32	2e-5	6
CR	✓	✗	8	2e-5	20
BCN	✓	✓	8	2e-5	15
CCN	✓	✓	16	2e-5	17
CNG	✓	✓	32	2e-5	20
CNSp	✓	✗	16	2e-5	5
CNSu	✓	✓	32	2e-5	15
CNI	✓	✗	16	2e-5	19
CNO	✓	✓	16	3e-4	12
CNSt	✓	✗	32	2e-5	16
HSO	✓	✓	32	3e-4	1
SRC	✓	✓	4	2e-5	15
SRE	✓	✓	32	2e-5	8
SRPl	✓	✓	8	2e-5	1
SRPr	✓	✗	4	2e-5	4
SRS	✓	✗	32	2e-5	4
DIA	✓	✗	8	2e-5	14
DII	✓	✗	32	2e-5	18

Table 4: Hyper-paramaters used for our models.

Table 4 displays the hyper-parameters used in this work. “Label input” refers to labeling the elements of each instance in the input, as demonstrated in Table 5, for example for AG.

B Task Instructions

Table 5 presents the natural language instructions used for each task, along with an example for each task.

Task	Instruction	Example
AH	Determine how helpful the advice is given a situation and advice.	Was summoned for Jury Duty in a state that I no longer live in and they won't approve my exclusion requests. [...] I'm pretty sure I was summoned because my Driver's License is still issued (MO), but I live in VA now. I don't have the means to get back to MO before the jurors are decided, but I also don't want to be held in contempt of court. Anything would be helpful, thanks! <sep> Well, you are long-overdue to get your VA Driver's License (you only have 60 days), so I suggest you do so now.
AG	Determine the grammaticality of an answer given a narrative, a question and an answer to that question.	narrative <sep> Tim was unemployed. He applied to many jobs. He interviewed for a car valet job. He got the job. He happily began working as a car valet. <sep> question <sep> Why did He get the job? <sep> answer <sep> Tim got the job because he impressed the company.
AV	Determine the plausibility of the answer given a narrative, a question and an answer to that question.	
QA	Determine the comprehensibility of the question and whether the narrative contains the answer for the question given a narrative and a question.	
HS	Determine how appropriate the follow up sentence is for the context given a context and a follow up sentence.	Then, the man writes over the snow covering the window of a car, and a woman wearing winter clothes smiles. then <sep> , the man continues removing the snow on his car.
CR	Given a concept set, a reference sentence, and a sentence, determine the plausibility of the sentence.	food eat chair sit <sep> A man sitting on a chair eating food. <sep> a man eats food and eat chair sit in the beach.
BCN	Given a hate speech, the target of the hate speech, and a counter narrative in response to the hate speech, determine how good the counter narrative is.	hate speech <sep> blind people don't know what's going on <sep> target <sep> DISABLED <sep> counter narrative <sep> The problem is that blind people are not aware of what is going on, they just want to be accepted as human beings.
CCN	Given a hate speech, the target of the hate speech, and a counter narrative in response to the hate speech, determine how appropriate the counter narrative is in a real case scenario.	
CNG	You are given a hate speech, the target of the hate speech, and a counter narrative in response to the hate speech. Determine the grammaticality of the counter narrative.	
CNSp	Determine how specific the arguments brought by the counter narrative are in response to the hate speech given a hate speech, the target of the hate speech, and a counter narrative in response to the hate speech.	
CNSu	Given a hate speech, the target of the hate speech, and a counter narrative in response to the hate speech, determine how suitable the counter narrative is to the hate speech in terms of semantic relatedness and in terms of not spreading hate.	
CNI	Determine how informative the counter narrative is given a hate speech and a counter narrative in response to the hate speech.	Some races have lower physical and cognitive abilities, the sooner we accept this, the better it will be for humanity. <sep> This is a racist statement that is not supported by science.
CNO	Given a hate speech and a counter narrative in response to the hate speech, determine how offensive the counter narrative is.	
CNSt	Evaluate the stance of the counter narrative given a hate speech and a counter narrative in response to the hate speech.	
HSO	You are given a hate speech. Determine how offensive the hate speech is.	hate speech <sep> According to a recent OFSTED report, a school in Birmingham is still segregating girls and boys, despite a Hight Court ruling in 2017 that this is unlawful.
SRC	You have a story that consists of a premise, a second sentence, and an ending. A second story with the same premise is provided but the second sentence is replaced with a counterfactual sentence, and the ending is rewritten, determine how reasonable the rewritten ending is given the premise and counterfactual sentence provided.	story <sep> Wallace had a passion for gardening. Wallace never became a professional gardener. <sep> ending <sep> One day, he was approached by a professional gardener. The professional gardener suggested that they work together. Now, Wallace has a lot of money that he loves to pay back.
SRE	You have a story that consists of a premise, a second sentence, and an ending. A second story with the same premise is provided but the second sentence is replaced with a counterfactual sentence, and the ending is rewritten. Determine how reasonably the rewritten ending outlines a sequence of events.	
SRPl	You are given a story, an original ending and a rewritten ending. Determine how well the plot in the rewritten ending relates to the plot of the original ending.	
SRS	Determine how well the rewritten ending keeps in mind the details provided in the counterfactual given a story that consists of a premise, a second sentence, and an ending as well as a second story with the same premise is provided but the second sentence is replaced with a counterfactual sentence, and the ending is rewritten.	
SRPr	Determine how well the rewritten ending keeps in mind the details provided in the premise given a story that consists of a premise, a second sentence, and an ending as well as a second story with the same premise is provided but the second sentence is replaced with a counterfactual sentence, and the ending is rewritten.	
DIA	You are given a premise, a hypothesis, and an update sentence. Determine how much the much the update sentence weakens the hypothesis.	A girl in a black sweater and jeans pours water into an empty soda bottle. <sep> A girl pours water into an empty coca cola bottle <sep> The bottle is empty
DII	Given a premise, a hypothesis, and an update sentence, determine how much the much the update sentence strengthens the hypothesis.	A group of mountain climbers rests at the summit. <sep> A group of climbers celebrates at the top of Everest. <sep> The climbers are smiling

Table 5: Natural language instructions used for each task alongside data samples.