Defining and Detecting Vulnerability in Human Evaluation Guidelines: A Preliminary Study Towards Reliable NLG Evaluation

Jie Ruan*, **Wenqing Wang***, **Xiaojun Wan**
Peking University
{ruanjie,wangwenqing}@stu.pku.edu.cn
wanxiaojun@pku.edu.cn

Abstract

Human evaluation serves as the gold standard for assessing the quality of Natural Language Generation (NLG) systems. Nevertheless, the evaluation guideline, as a pivotal element ensuring reliable and reproducible human assessment, has received limited attention. Our investigation revealed that only 29.84% of recent papers involving human evaluation at top conferences release their evaluation guidelines, with vulnerabilities identified in 77.09% of these guidelines. Unreliable evaluation guidelines can yield inaccurate assessment outcomes, potentially impeding the advancement of NLG in the right direction. To address these challenges, we take an initial step towards reliable evaluation guidelines and propose the first human evaluation guideline dataset by collecting annotations of guidelines extracted from existing papers as well as generated via Large Language Models (LLMs). We then introduce a taxonomy of eight vulnerabilities and formulate a principle for composing evaluation guidelines. Furthermore, a method for detecting guideline vulnerabilities has been explored using LLMs, and we offer a set of recommendations to enhance reliability in human evaluation. The annotated human evaluation guideline dataset and code for the vulnerability detection method are publicly available online.[1]

1 Introduction

Natural Language Generation (NLG) has found extensive applications across diverse domains. Nevertheless, evaluating the quality of generated outputs has posed a longstanding and formidable challenge due to the inherent diversity of expressions capable of conveying the same meaning (Howcroft et al., 2020; Zhou et al., 2022). This abundance of possible variations complicates the development of automated evaluation methods (Novikova et al., 2017b; Reiter and Belz, 2009a), thus necessitating the reliance on human evaluation as the gold standard and regarding it as a more reliable evaluation method in NLG (Celikyilmaz et al., 2020; Gatt and Krahmer, 2018; Gkatzia and Mahamood, 2015b; Mellish and Dale, 1998; van der Lee et al., 2018).

However, the evaluation guidelines, which play a crucial role in ensuring the reliability of human evaluation, have not received adequate emphasis. The transparency issues inherent in human evaluation guidelines raise concerns regarding the validity and reproducibility of the evaluation results (Schoch et al., 2020). To investigate this issue, we conducted a study based on 3,233 papers that we crawled from ACL, EMNLP, and NAACL conferences in the last three years. Surprisingly, we indicate that only 29.84% of the papers involving human evaluation release their human evaluation guidelines. Human evaluation guidelines are crucially important for ensuring that human assessments are conducted reliably. However, when papers fail to release the evaluation guidelines, there is no guarantee of the reliability and reproducibility of their evaluation results. Moreover, our analysis of the guidelines released by these papers uncovered a significant concern: a striking 77.09% of the released guidelines exhibited noticeable vulnerabilities[2], which could potentially have a detrimental impact on the correctness of human evaluation outcomes (Schoch et al., 2020).

The ultimate goal of establishing reliable human evaluation guidelines comprises several essential steps, which include detecting potential vulnerabilities in the guidelines, identifying specific vulnerability types, marking the precise segments with vulnerabilities, providing modification suggestions, and finally correcting identified vulnerabilities in

*Equal contribution.

[1] https://github.com/EnablerRx/GuidelineVulnDetect

[2] In this paper, "vulnerability" carries the same meaning as "defect", indicating issues within evaluation guidelines that could potentially result in unreliable evaluation outcomes.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7965–7989
June 16-21, 2024 ©2024 Association for Computational Linguistics

the guidelines. In this paper, we conduct a preliminary study on defining and detecting vulnerabilities in human evaluation guidelines, marking an initial step towards reliable guidelines. Specifically, we first constructed a human evaluation guideline dataset by collecting annotations of guidelines extracted from existing papers as well as generated via LLMs. Based on the analysis of the guidelines, we identified eight main categories of vulnerabilities including Ethical Issues, Unconscious Bias, Ambiguous Definition, Unclear Rating, Edge Cases, Prior Knowledge, Inflexible Instructions, and Others. The guidelines with vulnerabilities can result in issues such as annotators being unclear about task requirements, misunderstanding specific scoring standards, or erroneously directing annotators to assign higher scores to particular systems, which leads to incorrect and irreproducible evaluation results. To detect these vulnerabilities, we explored several prompt strategies to evoke the capability of current LLMs in vulnerability detection for human evaluation guidelines, and recommend an LLM-based method employing Chain of Thought (CoT) strategies.

The main contribution of this paper is as follows: 1) We are the first to study vulnerabilities in human evaluation guidelines and release the first human evaluation guideline dataset with annotated vulnerabilities for advancing reliable human evaluation; 2) We analyze the existing human evaluation guidelines and introduce a taxonomy of eight vulnerabilities for evaluation guidelines; Furthermore, we establish a principle for writing a reliable human evaluation guideline; 3) We explore an LLM-based method for detecting guideline vulnerabilities. We recommend employing this method to assess the reliability of the guidelines before conducting human evaluations; 4) We present a set of recommendations designed to elevate the reliability of human evaluation by offering guidance on writing robust guidelines and identifying potential vulnerabilities.

2 Raw Guideline Dataset

Due to the lack of existing work related to human evaluation guideline assessment, we construct the first human evaluation guideline dataset through two methods: extracting from existing papers and generating from GPT3.5, referred to as the **authentic guidelines** and **synthetic guidelines**, respectively. Note that we collect and analyze synthetic guidelines because LLMs has been proved as powerful tools for synthetic data generation (Agrawal et al., 2022; Liu et al., 2022; Bitton et al., 2023).

2.1 Authentic Guidelines

The construction of authentic guidelines involves a three-step process: First, we crawled papers[3] on ACL, EMNLP, and NAACL conferences from 2020 to 2022 and obtained 3,233 raw data. Then, we filter the papers using two groups of keywords, and narrow down the paper set to 319. *Human evaluation* and *manual assessment* constitute the first group, using them to focus solely on the 677 papers related to the evaluation tasks, while *guideline*, *instruction*, *questionnaire*, *interface*, and *screenshot* are employed as keywords to identify papers potentially containing guideline sections. Finally, we manually filtered out papers specifically related to NLG tasks and extract 227 guidelines from ACL (111), EMNLP (62) and NAACL (54). Any guidelines presented as figures or charts were converted into textual formats. More Details of the collected data can be found in Appendix A.

2.2 Synthetic Guidelines

Constructing effective prompts for language models to perform NLP tasks is currently a highly researched topic (Schick and Schütze, 2021; Le Scao and Rush, 2021; Tam et al., 2021; Logan IV et al., 2022; Reynolds and McDonell, 2021). Inspired by Mishra et al. (2022), we design 5 prompts to guide GPT-3.5-Turbo in generating diverse guidelines, including raw prompt, raw prompt with evaluation aspects, structured prompt, structured prompt with evaluation aspects and structured prompt with evaluation aspects and constraints, as shown in Appendix B. For each prompt, we expand the dataset by incorporating 12 NLG tasks and 2 evaluation settings, along with alternately utilizing the keywords *instruction* and *guideline*. Consequently, we generated a total of 48 guidelines for each prompt (12 tasks $\times$ 2 settings $\times$ 2 keywords).

2.3 Data Statistics

Finally, we obtained 227 authentic guidelines extracted from existing papers, alongside 239[4] synthetic guidelines generated by GPT-3.5-Turbo with average lengths of 247.64 words and 237.05 words,

[3]We crawled https://paperswithcode.com, an open resource website which ensures our access to the guideline data once they are publicly available.

[4]A piece of synthetic guideline that doesn't belong to the evaluation task has been filtered out.

respectively. In total, our dataset comprises 466 human evaluation guidelines. It is worth noting that out of the 677 papers related to human evaluation, only 202 (29.84%) of them openly released their evaluation guidelines after considering cases where multiple guidelines were included in a single paper, indicating the insufficient attention given to the evaluation guidelines.

3 Guideline Vulnerability Annotation

3.1 Taxonomy of Guideline Vulnerability

We define a typology consisting of eight guideline vulnerabilities by analyzing the guidelines extracted from existing papers and generated by LLMs. An illustration for each type is shown in Table 1, which is designed for illustrative purposes and does not originate from the actual dataset. More examples can be found in Appendix C.

Ethical Issues (Mieskes, 2017): instructions do not consider potential ethical implications related to the evaluation process, like privacy, cultural sensitivity, accessibility, or the potential misuse of the evaluation results.

Unconscious Bias (Schoch et al., 2020): instructions unconsciously favors or disadvantages certain results.

Ambiguous Definition (Jurgens, 2014): instructions for task definition are unclear, vague, or imprecise that can be interpreted in multiple ways.

Unclear Rating (Amidei et al., 2019): instructions that lack standardized criteria for evaluating aspects or definition of each point on a rating scale, resulting in potential inconsistency in ratings.

Edge Cases (Ruggeri et al., 2023): instructions do not specify how to handle edge cases or exceptional situations that don't neatly fit into the usual categories or criteria.

Prior Knowledge (Sun et al., 2020): instructions assume that evaluators have certain background knowledge or familiarity with a specific subject matter, tool, or principle.

Inflexible Instructions: instructions are unnecessarily complex or rigid, making it hard for evaluators to follow and incapable of adjusting to variations in data or task requirements, which contradicts Sabou et al. (2014)'s conclusion that a simpler instruction tends to yield better results.

Finally, we add the additional type **Others** to ensure the completeness of the typology. This covers any vulnerabilities that do not fall into the above categories.

3.2 Data Annotation

We recruit four college students who possess English qualification certificates. Firstly, they were provided with an annotation guideline, which can be found in Appendix C. Each evaluator went through a training process (details in Appendix D) to enhance their understanding in the annotation process. Before annotation, we also designed a qualification test consisting of 10 guidelines, only annotators who passed the test were considered qualified and allowed to continue annotation. To ensure the annotation quality, we divided the dataset into batches and assigned the specific number of daily tasks to each annotator. Upon receiving the daily annotations, we reviewed the results and required the specific annotator to reannotate the batch of data assigned for that day if there was a low accuracy (less than 80%).

In the annotation interface, the authentic guidelines and synthetic guidelines are randomly presented on the left side so as to prevent bias, while the eight vulnerability types are displayed on the right. Annotators were instructed to assign the specific vulnerability types based on the predefined typology, or indicate "None" for guidelines where the vulnerability type is absent. Each sample was annotated by two distinct annotators and a third annotator made the final decision if they are in disagreement.

We utilized Cohen's kappa (Cohen, 1960) to measure the inter-annotator agreement and computed on a per-label basis so as to gain label-specific insights. Ultimately, we calculated the mean values across all labels to assess the overall agreement. The annotation process lasted approximately two weeks, culminating in a substantial inter-annotator agreement of Cohen's kappa with κ=0.722 on authentic guidelines and κ=0.737 on the synthetic guidelines. More annotation details can be found in Appendix D.

3.3 Annotation Result

Figure 1 reports the annotation results on both authentic and synthetic guidelines. While LLMs have shown impressive results in various generation tasks, its current capabilities to generate reliable evaluation guidelines is limited, with vulnerabilities over 50%. We also report the results of five prompts in Appendix B, indicating that structured instructions incorporating evaluation aspects exhibit the lowest vulnerability ratio. What is worth

Guideline for Opinion Summarization Quality Evaluation

Thank you for participating in this opinion summarization quality evaluation task!
Opinion summarization is the task of automatically generating summaries for a set of reviews about a specific target. In this task, we focus on movie reviews written by users from the Rotten Tomatoes website**[1]**. You will be presented with one human-written reference summary first along with three system summaries generated by trained neural networks respectively**[2]**. Please evaluate the quality of opinion summaries**[3]** with respect to the following four features: (1) Relevance; (2) Consistency; (3) Fluency; and (4) Coherence**[4]**. You should make comparisons for the summary evaluation and rank the four summaries in the order of the four evaluation aspects**[5]**, and the evaluation is conducted on the open-source annotation tool Doccano**[6]**.

IMPORTANT:

- Thoroughly read the guideline and familiarize yourself with the task of opinion summarization quality evaluation.
- Carefully read the source reviews as well as reference and system summaries to grasp the overall content.
- Evaluate the overall quality of each summary based on the four designated aspects, assign a score to each dimension sentence by sentence and aggregate all the scores of each sentence to perform pairwise comparisons**[7]**.
- If you encounter any difficulties or have questions during the annotation procedure, refer to the provided guidelines. Alternatively, feel free to contact us via email for further clarification.

Vulnerabilities in Guideline

1. Ethical Issues: guiding in this manner disregards the personal privacy of the commenters as it fails to specify whether the comments are anonymous or obtained with user consent. An improved guideline should address ethical concerns such as “All anonymized reviews have been previously collected with user consent and have been stripped of personally identifiable information.”
2. Unconscious Bias: guiding in this manner specifies the sequence of the summaries, leading evaluators to have a biased perception of the reference as superior in quality. An improved guideline should be more neutral such as “You will be presented with four summaries in a random order, including one reference summary and three system summaries generated by trained neural networks."
3. Ambiguous Definition: guiding in this manner fails to clarify whether the task is to evaluate four summaries based on the source review or to evaluate three system-generated summaries based on the reference. An improved guideline should provide a more explicit task definition such as “Please evaluate the quality of both the reference and three system-generated opinion summaries given the corresponding source review."
4. Unclear Rating: guiding in this manner lacks a clear explanation of the evaluation aspect, which leads to multiple interpretations for different evaluators, resulting in inconsistent ratings. Given that this task involves pairwise comparison, an improved guideline doesn’t have to provide a rating scale, yet it should explicit the evaluation criteria such as: “(1) Relevance: measures how well the summary captures the key points of the source review; (2) Consistency: measures whether the facts in the summary are consistent with the facts in the source review; (3) Fluency: measures the quality of individual sentences, are they well-written and grammatically correct; (4) Coherence: measures the quality of all sentences collectively, to the fit together and sound naturally."
5. Edge Cases: guiding in this manner fails to provide directions for addressing edge cases where both summaries have equal quality. An improved guideline should comprehensively consider exceptional situations such as: “In the case of two summaries are of equal quality, place them side by side in the same ranking."
6. Prior Knowledge: guiding in this manner assumes evaluators have annotation experience without explaining how to use the professional tool Doccano. An improved guideline should offer training or detailed explanations for professional tools and principles such as “The evaluation is conducted on the open-source annotation tool Doccan, and subsequently, training will be provided on how to use it for annotation. If you are interested, you can visit this website in advance for more information: https://doccano.github.io/doccano."
7. Inflexible Instructions: guiding in this manner makes the task unnecessarily complex by aggregating individual sentences for overall quality evaluation. Furthermore, it doesn’t align with certain aspects, such as coherence, which require an evaluation that considers all sentences collectively. An improved guideline should be more flexible and reasonable such as “Evaluate the overall quality of each summary and make comparisons based on the four designated aspects."

Table 1: An illustration of the taxonomy on guideline vulnerability types.

noting is that the quality of the authentic guidelines extracted from existing papers is much poorer, and the vulnerability ratio is 77.09%, which undermines the reliability of evaluation tasks. This aligns with the conclusions of Sabou et al. (2014), who demonstrated that the crowdsourcing community still lacks a set of best-practice guidelines, resulting in low-quality annotations. We make a call for future researchers to be aware of the issue and emphasize the need for thorough refinement and investigation to develop a robust guideline.

Regarding the vulnerability type, both the authentic and synthetic guidelines are in a similar distribution that Ambiguous Definition and Unclear Rating occur most frequently. The vulnerability "Others" appears in scenarios such as when guidelines generated by LLMs are incomplete. Table 2 shows the authentic guideline for the machine-in-the-loop writing of image caption task extracted from Padmakumar and He (2022). The guideline lacks the definition of the machine-in-the-loop writing task and fails to specify the evaluation criterion, leaving uncertainty about the annotation process. As a result, the reliability and validity of the evaluation process will be compromised. Apart from the two types, authentic guidelines exhibit more vulnerabilities of bias, prior knowledge and ethical issues. Additionally, authentic guidelines are

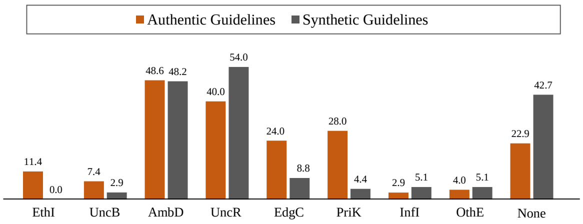

Figure 1: Distributions of vulnerability types on authentic and synthetic guidelines with EthI, UncB, AmbD, UncR, EdgC, PriK, InfI, OthE refers to Ethical Issues, Unconscious Bias, Ambiguous Definition, Unclear Rating, Edge Cases, Prior Knowledge, Inflexible Instructions and Others respectively. "None" means the guideline has no vulnerability at all. The ratio calculation is achieved by taking the number of guidelines that include a particular category and dividing it by the total count of guidelines.

Instructions for crowdworkers evaluating the captions • Choose the better (more descriptive and/or figurative) caption for the image. • A better caption is your subjective judgement, the rubrics to make the choice are that the caption is descriptive and/or figurative in its interpretation of the image (Refer the examples for further clarification). • The explanation asked is supposed to be very brief. A single word of if you like it for being descriptive or interpretive will do. • Relevance of the caption to the image is your subjective choice whether the caption appropriately represents what is in the image and is not just a catchy piece of text unrelated to the image. • A caption that you deem irrelevant should never be the better caption, unless both are irrelevant.

Table 2: Authentic human evaluation guideline extracted from Padmakumar and He (2022) with vulnerabilities of Ambiguous Definition and Unclear Rating.

more likely to suffer from neglecting edge cases, whereas the LLM is more prone to generate excessively rigid and complex guidelines, resulting in more vulnerabilities of Inflexible Instructions. As such, resorting to the LLMs to fill in the gaps proves to be a promising approach. However, it is important to note that the current LLM can only generate preliminary drafts of guidelines and needs more effective strategies to enhance reasoning ability for improving the reliability of guidelines. A future direction is to enhance LLM's reasoning ability to improve its capability in generating reliable guidelines.

4 Experiments

In this section, we investigate utilizing LLMs to detect the specific vulnerability types in each evaluation guideline, which is taken as a multi-label vulnerability type classification task.

4.1 Large Language Models

We perform our experiments utilizing both open-source and closed-source LLMs. For open-source models, we fine-tuned LLaMA-7B, an efficient and popular foundation language model with LoRA[5]. Additionally, we also experimented with Flan-T5-XXL[6], Flan-Alpaca-L[7], and Falcon-7B[8], respectively. For closed-source models, we select two widely accessible large language models: TEXT-DAVINCI-003[9] and GPT-3.5-turbo[10]. TEXT-DAVINCI-003 is developed using a combination of supervised instruction tuning and Reinforcement Learning from Human Feedback methodologies. GPT-3.5-Turbo is an enhanced version of the GPT-3 language model with instruction fine-tuning.

4.2 Prompting Strategies

Our exploration involves designing prompts for both zero-shot and few-shot scenarios, encompassing four distinct prompt templates ("Basic", "VDesc", "CoT-Basic" and "CoT-VDesc") under each scenario, thus yielding a total of eight prompts. Basic prompt offers only the name of the vulnerability type, whereas VDesc prompt expands on this by including definition for each type. Additionally, we investigate the Chain-of-Thought (CoT) prompting technique on both prompt templates. Detailed prompting design and the full prompts are detailed in Appendix E.

4.3 Baselines

We further implement and finetune three Transformer-based classifiers as baselines: BERT (Devlin et al., 2019) along with its successors XLNet (Yang et al., 2019) and ALBERT (Lan et al., 2019), which have shown excellent performance on classification tasks. They are all deep pretrained models that first encodes a guideline into vector space by capturing contextual information bidirectionally and then outputs the probability for each label independently. We finetune all the models on the base version and the hyper-parameters can be found in Appendix F.

[5]https://github.com/Lightning-AI/lit-llama
[6]https://huggingface.co/google/flan-t5-xxl
[7]https://huggingface.co/declare-lab/flan-alpaca-large
[8]https://huggingface.co/tiiuae/falcon-7b
[9]https://platform.openai.com/docs/models/gpt-3-5
[10]https://platform.openai.com/docs/models/gpt-3-5

Human Evaluation Guideline	LLM	LLM Answer
Read the following pairs of texts (source and candidate) and provide ratings between 0 to 100 based on the degree of similarity in **meaning** and preservation of **grammar** for each candidate text. Drag or click on the appropriate portion of the slider to provide a rating for each candidate text. Please refer to the scoring example before starting the first task. Note: There will be a candidate text in each HIT which contains several random words appearing out of context unrelated to the text. Please make sure to score this candidate text in each HIT with **10-20** points. Apart from this, please refer to the table below for scoring the candidate text. You must provide a rating for all examples to proceed. **Category: Meaning** Mild (1 sent): deduct **5-10** pts e.g., wrong pronoun Significant (1 sent): deduct **30** pts e.g., relevant words but diff meaning Mild (2 sent): deduct **10-20** pts Significant (2 sent): deduct **60-70** pts **Category: Grammaticality** Mild (1 sent): deduct **5-10** pts e.g., missing conjunction Significant (1 sent): deduct **25** pts e.g., incomplete sentence Mild (2 sent): deduct **10-20** pts Significant (2 sent): deduct **50-60** pts	**GOLD**	Ambiguous Definition, Unclear Rating
	TEXT-DAVINCI-003	REASONING: The guideline does not provide a clear definition of the task, making it difficult for evaluators to interpret the instructions. It also does not provide a detailed explanation of the rating aspects, leading to multiple interpretations for different evaluators, causing "Unclear Rating". So the answer is "Ambiguous Definition" and "Unclear Rating". ANSWER: Ambiguous Definition, Unclear Rating ✓
	GPT-3.5-Turbo	REASONING: The guideline does not provide clear instructions on how to handle edge cases where the candidate text contains random words unrelated to the text. So the answer is "Edge Cases" ANSWER: Edge Cases ✗
	LLaMA	Mild, Significant ✗
	Flan	Let's think step by step: REASONING: The first task is to score ✗
	Flan-Alpaca	Categoryeaningequal to 12 pts e ✗
	Falcon	None ✗

Table 3: Example answers of vulnerability types detected (generated) by LLMs with few-shot CoT prompting for a human evaluation guideline in the test set.

4.4 Data Splits

We initially divide the dataset into five parts, with four parts designated for training (80%) and one for testing (20%). The training set is used for supervised fine-tuning of pretrained baselines and is subsequently divided into train/validation sets in a 4:1 ratio. Further, each of these five parts is used as an individual testing set, while the remaining four parts serve as training sets. As such, we evaluate the performance of the baselines and LLMs across the entire dataset, treating each part as a test set in rotation, so as to mitigate random fluctuations due to the relatively small size of the dataset and obtain a more accurate performance estimate.

4.5 Evaluation Metrics

Following Chen et al. (2017), we adopt **macro-Precision(macro-P)**, **macro-Recall (macro-R)**, and **macro-F1** scores (Vechtomova, 2009), which assess the overall performance of a classifier by taking the average of Precision, Recall, and F1-scores across all individual labels for each class (including "None"). Considering the unequal proportions of different vulnerability types in the dataset, as shown in Figure 1, macro metrics can provide a more balanced view of the model's performance across all classes, as opposed to micro-averaging (Vechtomova, 2009), which gives more weight to the larger classes. Furthermore, we follow Ganda and Buch (2018) and utilize **Accuracy (ACC)** to assess the average accuracy of all individual types. We also follow Wu and Zhou (2017) and use the **instance-AUC (AUC)** metric. **Hamming Loss** (Schapire and Singer, 1998) is also incorporated, which evaluates the fraction of misclassified instance-label pairs, accounting for both missed relevant labels and predicted irrelevant labels.

5 Results and Analysis

5.1 Qualitative Analysis

We first show the case study of a sample from the test set in Table 3, in which the authentic guideline is drawn from Kim et al. (2021) and suffers from vulnerabilities of Ambiguous Definition and Unclear Rating. The answers are generated by LLMs under few-shot CoT prompting. We can find that TEXT-DAVINCI-003 not only generates completely correct answers, but also narrows down the scope of vulnerabilities in its reasoning, facilitating the correction of identified vulnerabilities in the guidelines. Nevertheless, GPT-3.5-Turbo appears to have misconstrued the definition of the Edge Cases, since the handling of cases "where the candidate text contains random words unrelated to the text" has already been provided. The four open-source models, on the other hand, don't generate the answer as instructed. Instead, LLaMA extracts keywords directly as the output, Flan and Flan-Alpaca yields nonsensical results, and Falcon consistently outputs “None” for all of the data, revealing the inefficiency of open-source models in vulnerability detection. We speculate the reason is due to the limited training data and the excessive length of the instructions.

5.2 Quantitative Analysis

Given that the results generated by open-source LLMs are invalid, as demonstrated in Table 3, quantitative evaluation becomes unfeasible. Therefore, we focus on GPT models and pre-trained baselines for quantitative analysis. Table 4 shows the experiment results for guideline vulnerability detection on both authentic and synthetic guidelines along with the entire dataset. We also report the results of each vulnerability type in Appendix G.

We first explored the effects of different prompt strategies, including Basic, Vdesc, and the use of CoT. Subsequently, we explored the detection performance of LLMs in zero-shot and few-shot settings. Additionally, we investigated the performance of different LLMs, namely TEXT-DAVINCI-003, GPT-3.5-Turbo, and pre-trained models including BERT, XLNet, and ALBERT. Finally, we analyzed the varying performance between authentic guidelines and synthetic guidelines. Through this exploration of different prompt strategies, models, and settings, we conclude that TEXT-DAVINCI-003 demonstrates superior performance with few-shot prompting and CoT strategies. Our analysis of experimental results exploring different prompt strategies, models, and settings is based on a comprehensive consideration of all evaluation metrics. When drawing conclusions from specific metrics, we specify the particular metrics that serve as the basis for our conclusions.

Regarding the Basic and VDesc prompt templates, they exhibit comparable capabilities. The reason is that the incorporation of vulnerability descriptions might potentially disrupt the reasoning process of LLMs, although they might provide detailed vulnerability descriptions for LLMs. According to results on All guidelines, we can also find that CoT generally improves model performance in all prompt strategies of zero-shot setting and Vdesc prompt strategy of few-shot setting. The reason why CoT doesn’t consistently enhance model performance in the Basic prompt strategy may stem from the insufficiency of vulnerability information provided by the Basic prompt for effective reasoning. For few-shot and zero-shot settings, we can conclude from the results on All guidelines that LLMs generally exhibit enhanced performance in few-shot scenarios.

In the analysis of various LLMs and pretrained models, the experimental results indicate that TEXT-DAVINCI-003 and GPT-3.5-Turbo exhibit comparable performance, consistently outperforming pretrained models across the majority of prompt strategies. However, the pretrained models still serve as robust baselines, showing specific advantages over TEXT-DAVINCI-003 without CoT strategies under zero-shot scenarios. A noteworthy observation is that Recall values generally surpass Precision in most cases, indicating a tendency for the models to classify guidelines as positive, i.e., no vulnerability is detected. Furthermore, observing the results of each vulnerability type (detailed experimental results are shown in Appendix G), it is found that the model’s ability to detect different vulnerabilities varies significantly. All these gaps suggest that the models still have room for improvement in guideline vulnerability detection.

We also compare the model’s performance on the two categories of guidelines: Authentic Guidelines and Synthetic Guidelines. Experimental results in Table 4 indicate that LLMs exhibit a stronger ability to detect vulnerabilities in synthetic guidelines compared to authentic guidelines. Moreover, TEXT-DAVINCI-003 with CoT-Vdesc strategy demonstrates superior detection capabilities in detecting authentic guidelines, while GPT-3.5-Turbo with CoT-Vdesc strategy exhibits enhanced detection proficiency for synthetic guidelines. Overall, the experimental results show TEXT-DAVINCI-003 exhibits superior detection capabilities in detecting all guidelines.

Based on the outcomes of a thorough exploration involving various prompt strategies, models, and settings, our conclusion is that TEXT-DAVINCI-003 demonstrates superior performance with few-shot prompting and CoT strategies. Overall, TEXT-DAVINCI-003 with CoT-Vdesc prompt strategy in the few-shot scenario has the best performance for all guidelines and is recommended as the method for guideline vulnerability detection.

6 Practical Recommendations

We summarize the key findings from this work and provide practical recommendations for reliable

Model	Prompt	Macro-P			Macro-R			Macro-F1			ACC			AUC			Hamming Loss↓		
		Aut	Syn	All	Aut	Syn	All	Aut	Syn	All	Aut	Syn	All	Aut	Syn	All	Aut	Syn	All
LLMs zero-shot																			
TEXT-DAVINCI-003	Basic	0.43	0.49	0.47	0.52	0.49	0.50	0.41	0.46	0.44	0.69	0.72	0.71	0.42	0.53	0.49	0.31	0.28	0.29
	Vdesc	0.43	0.47	0.46	0.45	0.52	0.49	0.44	0.49	0.47	0.76	0.81	0.79	0.45	0.53	0.50	0.21	0.14	**0.17**
	CoT-Basic	0.49	0.49	0.49	0.52	0.51	0.51	0.36	0.41	0.39	0.54	0.59	0.57	0.46	0.57	0.53	0.46	0.41	0.43
	CoT-Vdesc	0.46	0.52	0.50	0.47	0.54	0.51	0.39	0.46	0.43	0.60	0.66	0.64	0.45	0.60	0.54	0.40	0.34	0.36
GPT-3.5-Turbo	Basic	0.53	0.48	0.50	0.58	0.53	0.55	0.48	0.41	0.44	0.64	0.61	0.63	0.60	0.39	0.55	0.36	0.51	0.37
	Vdesc	0.53	0.53	0.53	**0.61**	0.55	0.58	0.44	0.48	0.46	0.56	0.69	0.64	0.63	0.60	0.61	0.44	0.31	0.36
	CoT-Basic	0.53	0.52	0.53	0.58	0.55	0.56	0.50	0.52	0.51	0.67	0.76	0.72	0.59	0.59	0.59	0.33	0.24	0.28
	CoT-Vdesc	0.51	0.53	0.53	0.57	0.55	0.56	0.49	0.53	0.51	0.69	0.81	0.76	0.55	0.58	0.57	0.31	0.19	0.24
LLMs few-shot																			
TEXT-DAVINCI-003	Basic	0.51	0.55	0.53	0.58	0.51	0.54	0.51	0.50	0.50	0.78	0.76	0.77	0.54	0.54	0.54	0.22	0.24	0.23
	Vdesc	0.48	0.56	0.53	0.52	0.55	0.54	0.43	0.52	0.49	0.73	0.79	0.77	0.50	0.61	0.57	0.27	0.21	0.23
	CoT-Basic	**0.54**	0.47	0.50	0.57	0.46	0.51	0.53	0.46	0.49	0.79	0.80	0.79	0.65	0.59	0.62	0.21	0.20	0.21
	CoT-Vdesc	**0.54**	0.68	**0.61**	0.58	0.61	**0.60**	**0.54**	0.62	**0.58**	**0.80**	**0.87**	**0.83**	**0.67**	**0.69**	**0.68**	**0.20**	**0.13**	**0.17**
GPT-3.5-Turbo	Basic	0.44	0.60	0.51	0.48	0.68	0.58	0.46	0.56	0.51	0.71	0.85	0.78	0.48	0.60	0.54	0.29	0.15	0.22
	Vdesc	0.45	0.60	0.52	0.48	**0.73**	**0.60**	0.46	0.58	0.52	0.71	0.85	0.78	0.49	0.61	0.55	0.29	0.15	0.22
	CoT-Basic	0.46	0.55	0.51	0.49	0.57	0.54	0.46	0.54	0.51	0.74	0.85	0.79	0.50	0.61	0.56	0.26	0.15	0.21
	CoT-Vdesc	0.44	**0.71**	0.60	0.50	0.68	**0.60**	0.46	**0.67**	**0.58**	0.75	**0.87**	0.80	0.50	0.66	0.59	0.24	**0.13**	0.20
Baseline																			
BERT		0.48	0.50	0.49	0.48	0.50	0.49	0.48	0.47	0.48	0.75	0.81	0.79	0.52	0.53	0.53	0.25	0.19	0.21
XLNet		0.45	0.48	0.47	0.48	0.49	0.49	0.48	0.47	0.47	0.76	0.80	0.79	0.56	0.52	0.53	0.24	0.20	0.21
ALBERT		0.45	0.49	0.47	0.48	0.50	0.49	0.45	0.48	0.46	0.79	0.80	0.80	0.45	0.61	0.51	**0.20**	0.21	0.20

Table 4: Guideline vulnerability detection results on "Authentic Guidelines (Aut)", "Synthetic Guidelines (Syn)" and the whole dataset (All). Upper, middle and lower parts show results of LLMs under zero-shot and few-shot scenarios as well as baseline models, respectively. The best values of each column are bolded. ↓ indicates that the lower value indicates the better performance.

human evaluation.

1. Writing human evaluation guidelines using LLMs. Our research has found that the proportion of vulnerabilities in guidelines generated by LLMs is lower than those written by humans. We suggest directly instructing LLMs about the requirements for evaluation and utilizing them to generate human evaluation guidelines.

2. Modify the evaluation guideline draft written by LLMs based on the proposed principles for human evaluation guidelines (shown in Appendix H). We analyze human evaluation guidelines and summarize principles to compose a robust evaluation guidelines. We recommend referencing these principles when crafting the guidelines.

3. Utilize TEXT-DAVINCI-003 with CoT-VDesc strategy to identify vulnerabilities. It has been proven to be an efficient, convenient, and cost-effective method, and detecting a guideline only requires approximately \$0.02[11].

4. Conduct human evaluation in strict accordance with the human evaluation guidelines.

5. Publicly release the human evaluation guideline. This can contribute to the transparency of human evaluation.

7 Related Work

7.1 Vulnerability Detection

Early explorations in Vulnerability Detection (VD) span rule-based approaches targeting predefined patterns. Subsequent advancements incorporate Machine Learning (ML) and Deep Learning (DL) techniques to predict vulnerabilities automatically in various tasks, including software security detection (Li et al., 2018), smart contract opcodes detection (Qian et al., 2022) and code vulnerability detection (Cheshkov et al., 2023). Recently, inspired by the outstanding performance of LLM in code-based tasks, Cheshkov et al. (2023) attempted to explore the capability of LLM in addressing code vulnerability detection. Vulnerability detection already being implemented across a wide range of tasks with various techniques, yet none of them

[11]The prompt consists of 909 tokens, with the inclusion of the average length of each guideline(242.21 tokens), multiplied by the cost of TEXT-DAVINCI-003 (\$0.0200 / 1K).

have been designed to explore the issue of vulnerability detection in human evaluation guidelines.

7.2 Natural Language Generation Evaluation

Previous studies have frequently relied on automatic metrics like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), BERT-SCORE (Zhang et al., 2019), MOVER-SCORE (Zhao et al., 2019) and BART-SCORE (Yuan et al., 2021) to evaluate the quality of generated text, primarily due to their cost-effectiveness, quickness, and repeatability (Reiter and Belz, 2009b). Nevertheless, these metrics have been criticized for their limited interpretability (van der Lee et al., 2019) and low correlation with human judgements (Belz and Reiter, 2006; Liu et al., 2016; Reiter and Belz, 2009b; Novikova et al., 2017a). Human evaluation is widely recognized as the gold standard for evaluating NLG systems (Mellish and Dale, 1998; Gkatzia and Mahamood, 2015a; van der Lee et al., 2018). However, it has the potential to be unreliable due to cognitive biases (Schoch et al., 2020) and the lack of standardized evaluation methodologies (van der Lee et al., 2019). Shimorina and Belz (2021) contributes to transparency in the human evaluation process by documenting it, while Belz et al. (2023) explores reproducibility in NLP human evaluation. However, there is currently no comprehensive work addressing the reliability of human evaluation guidelines, a pivotal element ensuring reliable and reproducible human assessment. With the increasing interest in LLMs, recent studies have been conducted to examine their suitability for assessing generation tasks, like summarization (Luo et al., 2023), machine translation (Kocmi and Federmann, 2023), etc. In this work, we focus on both human evaluation and large language model evaluation, which is the first to utilize LLMs for assessing guidelines in human evaluation.

8 Conclusion

In this paper, we propose and analyze significant issues in the evaluation guidelines of gold-standard human assessments. We conduct a preliminary study on defining and detecting vulnerabilities in guidelines to advancing reliable human evaluation. By proposing a taxonomy of guideline vulnerabilities, we constructed the first annotated human evaluation guideline dataset. We then explored LLMs with Few-Shot prompting and CoT strategies for automatic vulnerability detection. Recommendations include employing LLMs to assist in writing human evaluation guidelines and modifying them based on the proposed principles. Utilizing the proposed LLM-based vulnerability detection method is suggested for assessing the reliability of the guidelines.

In future work, we will delve into the precise annotation of spans containing vulnerabilities in guidelines, providing correction suggestions, automatically correcting identified vulnerabilities in the guidelines, and generating reliable guidelines by AI models. This advancement aims to contribute towards the ultimate goal of establishing dependable gold-standard human evaluation guidelines, thereby enhancing the reliability of NLG assessments.

Acknowledgements

This work was supported by Beijing Science and Technology Program (Z231100007423011), National Key R&D Program of China (2021YFF0901502), National Science Foundation of China (No. 62161160339) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

Limitations

This study serves as a preliminary exploration towards establishing reliable evaluation guidelines. We proposed and analyzed significant issues in gold-standard human assessments, specifically focusing on identifying vulnerabilities in guidelines. Our preliminary study employed LLMs to detect guideline vulnerabilities and provided recommendations for improving reliability in human evaluation. However, the ultimate goal of achieving dependable gold-standard human evaluation guidelines requires further investigation. Future work can delve into precise annotation of spans containing vulnerabilities, automatic correction of identified issues, and the generation of reliable guidelines using AI models. These advancements aim to contribute to establishing dependable guidelines, thereby enhancing the reliability of NLG assessments. It is important to note that due to cost considerations, experiments with the proposed method were not conducted on GPT-4. Implementing the proposed method on GPT-4 may further enhance its effectiveness, a consideration for future research.

Ethics Statement

We recruit annotators from a college campus. They are completely free to decide whether or not to participate in our annotation. The payment is 9 dollars per hour, higher than the local minimum wage. There is no personal information in our collected dataset. The information which may be used to identify the participants is deleted after the annotation. Moreover, the LLM-generated guidelines may contain toxic language, which can make annotators uncomfortable. We reviewed the data before annotation and found no problematic samples. We check the licenses of the artifacts used in this study and do not find conflicts. The license of the dataset we will release is CC BY-NC 4.0.

References

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2022. Qameleon: Multilingual qa with only 5 examples. *arXiv preprint arXiv:2211.08264*.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. The use of rating and Likert scales in natural language generation human evaluation tasks: A review and some recommendations. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 397–402, Tokyo, Japan. Association for Computational Linguistics.

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.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320, Trento, Italy. Association for Computational Linguistics.

Anya Belz, Craig Thomson, Ehud Reiter, Gavin Abercrombie, Jose M Alonso-Moral, Mohammad Arvan, Jackie Cheung, Mark Cieliebak, Elizabeth Clark, Kees van Deemter, et al. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in nlp. *arXiv preprint arXiv:2305.01633*.

Yonatan Bitton, Shlomi Cohen-Ganor, Ido Hakimi, Yoad Lewenberg, Roee Aharoni, and Enav Weinreb. 2023. q2d: Turning questions into dialogs to teach models how to search. *arXiv preprint arXiv:2304.14318*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International joint conference on neural networks (IJCNN)*, pages 2377–2383. IEEE.

Anton Cheshkov, Pavel Zadorozhny, and Rodion Levichev. 2023. Evaluation of chatgpt model for vulnerability detection. *arXiv preprint arXiv:2304.07232*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhatri Ganda and Rachana Buch. 2018. A survey on multi label classification. *Recent Trends in Programming Languages*, 5(1):19–23.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Dimitra Gkatzia and Saad Mahamood. 2015a. A snapshot of NLG evaluation practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60, Brighton, UK. Association for Computational Linguistics.

Dimitra Gkatzia and Saad Mahamood. 2015b. A snapshot of nlg evaluation practices 2005-2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60.

David M Howcroft, Anja Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel Van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3006–3012, Reykjavik, Iceland. European Language Resources Association (ELRA).

Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. 2021. BiSECT: Learning to split and rephrase sentences with bitexts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6193–6209, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Zhen Li, Deqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, and Yuyi Zhong. 2018. Vuldeepecker: A deep learning-based system for vulnerability detection. *arXiv preprint arXiv:1801.01681*.

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.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. 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.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for abstractive text summarization.

C Mellish and R Dale. 1998. Evaluation in the context of natural language generation. *Computer Speech amp; Language*, 12(4):349–373.

Margot Mieskes. 2017. A quantitative study of data in the NLP community. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 23–29, Valencia, Spain. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017a. 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.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017b. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*.

Vishakh Padmakumar and He He. 2022. Machine-in-the-loop rewriting for creative image captioning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 573–586, Seattle, United States. Association for Computational Linguistics.

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.

Shenyi Qian, Haohan Ning, Yaqiong He, and Mengqi Chen. 2022. Multi-label vulnerability detection of smart contracts based on bi-lstm and attention mechanism. *Electronics*, 11(19):3260.

Ehud Reiter and Anja Belz. 2009a. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Ehud Reiter and Anja Belz. 2009b. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Shamik Roy, Maria Leonor Pacheco, and Dan Goldwasser. 2021. Identifying morality frames in political tweets using relational learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9939–9958, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Federico Ruggeri, Francesco Antici, Andrea Galassi, Katerina Korre, Arianna Muti, and Alberto Barrón-Cedeño. 2023. On the definition of prescriptive annotation guidelines for language-agnostic subjectivity detection. In *Proceedings of Text2Story—Sixth Workshop on Narrative Extraction From Texts, held in conjunction with the 45th European Conference on Information Retrieval (ECIR 2023)*, volume 3370, pages 103–111. CEUR-WS. org.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland. European Language Resources Association (ELRA).

Robert E Schapire and Yoram Singer. 1998. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 80–91.

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. 2020. "this is a problem, don't you agree?" framing and bias in human evaluation for natural language generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 10–16, Online (Dublin, Ireland). Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Hila Gonen, Ari Holtzman, Yulia Tsvetkov, and Luke Zettlemoyer. 2022. Toward human readable prompt tuning: Kubrick's the shining is a good movie, and a good prompt too? *arXiv preprint arXiv:2212.10539*.

Anastasia Shimorina and Anya Belz. 2021. The human evaluation datasheet 1.0: A template for recording details of human evaluation experiments in nlp. *arXiv preprint arXiv:2103.09710*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging Chinese machine reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:141–155.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Chris van der Lee, Bart Verduijn, Emiel Krahmer, and Sander Wubben. 2018. Evaluating the text quality, human likeness and tailoring component of PASS: A Dutch data-to-text system for soccer. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 962–972, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Olga Vechtomova. 2009. Book review: Introduction to information retrieval by christopher D. manning, prabhakar raghavan, and hinrich Schütze. *Computational Linguistics*, 35(2).

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xi-Zhu Wu and Zhi-Hua Zhou. 2017. A unified view of multi-label performance measures. In *international conference on machine learning*, pages 3780–3788. PMLR.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

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

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

Kaitlyn Zhou, Su Lin Blodgett, Adam Trischler, Hal Daumé III, Kaheer Suleman, and Alexandra Olteanu. 2022. Deconstructing nlg evaluation: Evaluation practices, assumptions, and their implications. *arXiv preprint arXiv:2205.06828*.

A Authentic Guidelines Details

For the collected data, we focused on work related to human evaluation in NLG tasks. In descending order of frequency, specific tasks include summarization (42), dialogue generation(36), question answering (34), machine translation (26), story generation (20), image captioning (9), etc. These guidelines are collected from high-quality NLP conferences ACL, EMNLP and NAACL over the past three years (2020-2022). Apart from machine translation that cover a range of language pairs like English-French, English-Japanese, Chinese - English, English-German, English-Spanish, and English-Russian, most of the tasks primarily focus on the English language. Additionally, we have gathered information on the reported inter-annotator agreement, revealing a general inverse relationship between the number of identified vulnerabilities and the level of agreement. To illustrate, in the vulnerability-free guideline from Jiang et al. (2020), Cohen's Kappa can reach a substantial level of 0.807, whereas in the guideline from Roy et al. (2021), with three identified vulnerabilities, Cohen's Kappa falls only within the range of 0.50 to 0.64. The list of crawled papers and the guidelines with annotations are released.

B Prompts for Synthetic Guideline Generation

To explore the LLM's ability in writing human evaluation guidelines and extend the guideline datasets, we utilize different prompt strategies for LLMs to generate diverse human evaluation guidelines. Table 6 displays the prompts that were employed for creating synthetic guidelines, which fall into two categories: raw instructions and structured instructions. Inspired by the sensitivity of language models to the framing of their instructional prompts (Mishra et al., 2022), we explore the impact of incorporating evaluation aspects and constraints separately, with a total of five prompt variations. For each prompt, we analyze their performance across 12 NLG tasks: summarization, machine translation, dialogue generation, story generation, paraphrase generation, data to text, grammar error correction, text simplification, code generation, code summarization, question generation, and spelling correction, involve two assessment methods: direct assessment and pairwise comparison, and focus on the keywords "guideline" and "instruction".

The annotation result of each prompt can be found in Table 5. It can be seen that structured instructions, as opposed to raw instructions, generally contain fewer vulnerabilities and both can enhance generation performance after adding evaluation aspects. However, incorporating constraints into the prompt leads to a drop in generation performance, contradicting the findings of Shi et al. (2022), who employ a fluency constraint and observed an enhancement in performance.

Prompt	**% Vulnerability Ratio ↓**
raw	13.8
raw with aspect	10.5
structured	10.9
structured with aspect	9.6
structured with aspect and constraint	12.6

Table 5: Annotation results regarding the vulnerability ratio of each prompt variation. The ratio calculation involves dividing the count of synthetic guidelines containing vulnerabilities for a specific prompt by the overall guideline count. ↓ indicates a lower value is preferable.

C Annotation Guideline

We release our full guideline provided to crowd-worker participants for the manual evaluation of the vulnerability detection task in Table 7. We advocate for more related works to share their guidelines, aiming to enhance the transparency of human evaluation and thereby contribute to the establishment of a set of best-practice guidelines for the community.

D Annotation Details

The annotators we recruited are four college students with College English Test-6 certificates who are fluent in both English and Chinese languages, with Chinese as their mother tongue. There are 1 females and 3 males, with an average age of around 24. Then we conduct a training process. Specifically, we conducted an online meeting for annotator training, covering the interpretation of annotation guidelines, explanations and examples of various guideline vulnerabilities, clarification of relevant considerations, and a QA session. To confirm their proficiency, annotators underwent a pre-annotation test, and only those who passed were allowed to proceed with the formal annotation. Specifically, 10 guidelines are randomly sampled with 5 in Au-

	Five prompt variations for two instructions
RAW INSTRUCTION	**Raw prompt:** Write a human evaluation *guideline* for the *Summarization* task. The evaluation type is *Pairwise Comparison*. **Raw prompt with evaluation aspects:** Write a human evaluation *guideline* for the *Summarization* task. The evaluation type is *Pairwise Comparison*. Evaluate the following aspects: accuracy, coherence, consistency, relevance, fluency, informativeness, coverage, overall. The evaluation scale is 1-5 (1 is poor and 5 is excellent).
STRUCTURED INSTRUCTION	**Structured prompt:** Human evaluation task: *Summarization* Evaluation type: *Pairwise Comparison* Human evaluation *guideline*: **Structured prompt with evaluation aspects:** Human evaluation task: *Summarization* Evaluation type: *Pairwise Comparison* Evaluation aspects: accuracy, coherence, consistency, relevance, fluency, informativeness, coverage, overall Evaluation scale: 1-5 (1 is poor and 5 is excellent) Human evaluation *guideline*: **Structured prompt with evaluation aspects and constraints:** Human evaluation task: *Summarization* Evaluation type: *Pairwise Comparison* Evaluation aspects: accuracy, coherence, consistency, relevance, fluency, informativeness, coverage, overall Evaluation scale: 1-5 (1 is poor and 5 is excellent) Please be mindful of the following issues and avoid them: definition ambiguity, bias, assuming prior knowledge, insufficient coverage, lack of rating scale, lack of adaptability, and neglecting ethical implications Human evaluation *guideline*:

Table 6: Five prompts utilized for generating synthetic guidelines. The highlighted blue portions are employed across 12 NLG tasks, 2 evaluation methods and 2 key words, which are interchanged interchangeably.

thentic Guidelines and 5 in Synthetic Guidelines respectively. We annotated them first. Then, we calculated the accuracy of each participant based on our annotation. Higher accuracy indicates a more consistent understanding of our guidelines. Annotators who achieve at least 80% accuracy are considered qualified to continue the annotation.

We used Cohen's kappa(Cohen, 1960) to measure the inter-rater reliability. Considering that each label is independent and there are diverse label combinations for multi-label classification task, we do not require that two annotators provide completely identical label sets for each guideline. Instead, we assess the agreement between the two annotators in terms of each label they assign. Specifically, let n be the number of guidelines to be labeled by A and B two annotators. g is the number of distinct vulnerability labels, and f_{ij} denotes the frequency of the number of subjects with the i_{th} categorical response for annotator A and the j_{th} categorical response for annotator B. The kappa agreement is then calculated as:

$$p_0 = \frac{1}{n} \sum_{i=1}^{g} f_{ii},$$

$$p_e = \frac{1}{n^2} \sum_{i=1}^{g} f_{i+} f_{+i},$$

$$\kappa = \frac{p_0 - p_e}{1 - p_e},$$

where f_{i+} is the total for the i_{th} row f_{+i} and is the total for the i_{th} column in the frequency table.

E Prompt Template for Vulnerability Detection

The prompt templates used for vulnerability type detection on human evaluation guidelines are illustrated in Figure 2. As previously mentioned, we formulate four types of prompt templates: Basic, VDesc, CoT-Basic and CoT-VDesc in both zero-shot and few-shot scenarios. The zero-shot template of Basic template comprises of a Requirement + Constraints + Guideline framework. Requirement specifies the task motivation and remains consistent across all prompts; Constraints emphasize the desired output format such as *" Only reply with the names of vulnerabilities or 'None'"*; and Guideline represents the input data. On this basis, the template of VDesc further introduces the description of vulnerability types, which is expressed as Requirement + Description + Constraints + Guideline. We formulated the two prompt templates with the consideration that incorporating descriptions of vulnerability types can boost model effectiveness by offering more specific knowledge, yet simultaneously, it might also potentially disrupt the model's performance for introducing extra information.

Regarding the few-shot prompts, we expanded on the zero-shot method by incorporating seven pseudo-examples that encompass all vulnerability types except for "Others", some of which include multiple vulnerability types, so as to facilitate more appropriate model reasoning.

Additionally, we explore the CoT prompting technique, which elicits complex multi-step reasoning through step-by-step answer examples. For zero-shot prompts, we incorporate CoT by simply incorporating the phrase "Let's think step by step" before each answer, without supplying any examples (Kojima et al., 2022). It is worth noting that the phrase "Let's think step by step" is absent in the 7-shot prompt differing from zero-shot scenario. Instead, the phrase is integrated into the reasoning process of examples, through which we observed an improvement in performance. Inspired by Wang et al. (2022), we integrate the results of each reasoning over three runs and select the most consistent answer as the final answer set.

F Hyper-parameters

For TEXT-DAVINCI-003[12] and GPT-3.5-Turbo[13], the specific hyper-parameters can be found in Table 8. For the baselines, we utilize Huggingface[14] implementations for all the deep pretrained models and the specific hyper-parameters can be found in Table 9. We initially explored two approaches: one is to treat the multi-label classification task as a seq2seq problem, and generate a variable-length label sequence for a given text sequence. The other is to consider each neuron to be a binary classifier since the predictions for each category are independent in multi-label classification task, essentially forming a binary classification task for each label. We ultimately chose the second approach due to the limited training dataset in this task (less than 500 samples) and the increased data requirements of complex sequence models. We selected BCELoss as the loss function, which is commonly used in binary classification tasks. It calculates the individual losses for each label, quantifying the model's performance in terms of the difference between its predictions and the actual labels for each vulnerability type. Besides, we utilized the sigmoid activation function in the fully connected layers.

G Results of Each Vulnerability Type

Table 10, 11 and 12 report the experimental results of each vulnerability type (including "None") of pre-trained baselines, TEXT-DAVINCI-003 as well as GPT-3.5-Turbo respectively. Please note that the overall accuracy and AUC in Table 4 are the averages across eight types of vulnerabilities, which may vary with the inclusion of "None". We can observe that the model's capacity to detect different vulnerability types exhibits some variation, showing a trend in both LLMs and the Baseline, where more frequently occurring vulnerability types yield lower results. For LLMs, this is reasonable as they tend to output "None", as mentioned above, making them prone to misidentify more guidelines containing high-frequency vulnerability types as positive. Nevertheless, for the Baseline, which has undergone supervised training beforehand, we speculate this could be attributed to the limited size of the dataset, resulting in the models not having acquired robust capabilities yet. Additionally, the accuracy of "None" reaches highest at 0.79 in TEXT-DAVINCI-003 CoT-VDesc, which can be considered as an indicator of the reliability of the human evaluation guideline.

H Principal for Reliable Human Evaluation Guideline

A reliable human evaluation guideline is the beginning of reliable human evaluation. We provide the principal for composing a reliable human evaluation guideline in Table 13. For writing a reliable human evaluation guideline, researchers should provide explicit task definitions for raters and avoid biased instruction and prior knowledge assumptions. Moreover, the instruction should comprehensively cover a broad range of scenarios including the edge cases. Researchers should provide clear rating scale and criteria and make the instruction simple and flexible. Additionally, the potential ethical issues should be identified and addressed. It is highly recommended to attach examples and design a good user interface. Last but not least, remind annotators to be careful while carrying out their tasks to get more accurate evaluation results.

[12] https://platform.openai.com/docs/models/gpt-3-5
[13] https://platform.openai.com/docs/models/gpt-3-5
[14] https://huggingface.co/models

Vulnerability Detection in Human Evaluation Guidelines

Task Overview

Thank you for participating in this task! We are currently working on a project focused on crafting robust and reliable guidelines for human evaluation. You will be randomly presented with a human evaluation guideline extracted from existing papers or generated by Large Language Models (LLMs). Your job is to review the provided guidelines and identify potential vulnerabilities within the text. These vulnerabilities should fall into one or more of the eight categories outlined below.

Defect Types

Ethical Issues: instructions do not consider potential ethical implications related to the evaluation process, like privacy, cultural sensitivity, accessibility, or the potential misuse of the evaluation results.

- Ethical Issues: Evaluate the comments on this public social media post for sentiment analysis.
- Improved: Evaluate anonymized comments provided for sentiment analysis. All comments have been previously collected with user consent and have been stripped of personally identifiable information.

Unconscious Bias: instructions unconsciously favors or disadvantages certain results.

- Unconscious Bias: Evaluate the two systems A and B: How many points do you think system A is higher than system B?
- Improved: Evaluate the two systems A and B based on user satisfaction and score them respectively.

Ambiguous Definition: instructions for task definition are unclear, vague, or imprecise that can be interpreted in multiple ways.

- Ambiguous Definition:*Factual consistency of summarization* is defined as the accuracy and faithfulness of a summary in representing the source.
- Improved: *Factual consistency of summarization* is defined as the accuracy and faithfulness of a summary in representing the source. The *source* here usually has scenarios: the first is the input document and the second is common sense. In our task, we only focus on the first situation, i.e. evaluate the summary as factually inconsistent if it contains extra information of the input document,even though it is true facts.

Unclear Rating: instructions that lack standardized criteria for evaluating aspects or definition of each point on a rating scale, resulting in potential inconsistency in ratings.

- Unclear Rating: Rate the quality of the website.
- Improved: Rate the quality of the website based on its design, ease of navigation, and relevance of content on a scale of 1 to 5, where 1 is 'very poor' and 5 is 'excellent'. If the website was generally good but had one major flaw, consider rating it a 3or 4 depending on the severity of the flaw. If the website was poor but had one saving grace, consider rating it a 2 or 3.

Edge Cases: instructions do not specify how to handle edge cases or exceptional situations that don't neatly fit into the usual categories or criteria.

- Edge Cases: Evaluate the factuality error types in the summary, including: Hallucination Error, Entity Error, Particulars Error, Predicate Error, Coreference Error.
- Improved: Evaluate the factuality error types in the summary, including: Hallucination Error, Entity Error, Particulars Error, Predicate Error, Coreference Error. If the summary contains multiple errors, please list them all. If the error does not correspond to any of the above types, evaluate it as "Others".

Prior knowledge: instructions assume that evaluators have certain background knowledge or familiarity with a specific subject matter, tool, or principle.

- Prior knowledge: Evaluate the use of object-oriented programming (OOP) principles in the code.
- Improved: Evaluate the use of object-oriented programming (OOP) principles in the code. Check for the following aspects. If you are unfamiliar with these principles, please refer to https://baldur.gitbook.io/patters-and-best-practices/solid/oop-principles for more information."
  - Encapsulation:Object properties are hidden, and object properties need to be modified through object methods.
  - Inheritance:Subclasses can inherit the properties and methods of the parent class without redefining them.
  - Polymorphism: Polymorphism can be divided into static and dynamic. Static means that the same object can have different forms of expression, while dynamic means that a parent type can point to an instance of its subtype, making the subtype respond differently to the same method.
  - Abstraction:Abstraction refers to extracting the common attributes and behaviors of a class and storing them in a class, regardless of how the specific behaviors a rerealized.

Inflexible Instructions: instructions are unnecessarily complex or rigid, making it hard for evaluators to follow and incapable of adjusting to variations in data or task requirements.

- Inflexible Instructions:Evaluate the website's user interface design on a scale of 1 to 10, considering color aesthetics, balance between text and imagery, navigability, font choices,button placements, menu design, adherence to modern design principles, web page loading speed, and responsive design.
- Improved: Evaluate the website's user interface design on a scale of 1 to 10 from the perspectives of aesthetics, navigation and functionality.

Others: covers any vulnerabilities that do not fall into the above categories.

(Continued from previous page)

Annotation Procedure

- **Comprehension:** Carefully read through the entire human evaluation guideline in the center of the interface to get a full understanding of the content.
- **Labeling:** Identify and click all potential vulnerabilities within the guideline according to the eight defined categories: Definition Ambiguity, Bias, Assuming Prior Knowledge, Insufficient Coverage, Lack of Rating Scale, Lack of adaptability, Neglecting Ethical Implications and Others.
- **Review and Submit:** Repeat this process until the entire guideline has been thoroughly reviewed and all potential vulnerabilities have been identified. Press Enter to save and submit the annotation result.

Emphasis and Caution

- **Edge Cases:** Please note that a single guideline may contain more than one type of defect. In such scenarios,ensure to label all the appropriate defect types. If the defect does not fit any of the seven specific categories, classify it as "Others" and provide a brief explanation.
- **Daily Annotation Requirement:** The guideline for this task will be provided in batches. You are required to annotate a setof 30 items each day. Please submit the daily annotation results before 24:00(midnight) of that day.
- **Quality Assurance:** Each day, we will conduct a random inspection of the annotated data. If the accuracy rate falls below 80%, you will be required to re-annotate the data for that day. Please maintain high quality in your annotations.
- **Support and Reference:** If you encounter any confusion regarding professional knowledge or context while performing this task, please feel free to reach out to us for clarification.You may also refer to Wikipedia or other reliable sources to gain further understanding.
- **Feedback Mechanism:** Wehave set up a discussion board on the interface, where you can directly submit your queries, concerns, or suggestions through the button “click to comment on document”. This collaborative environment will allow for shared learning and problem-solving.

Examples of Defect Labelling

POSITIVE EXAMPLES

Guideline: Ensure you partake in a comprehensive interrogation of the quantifiable parameters that govern the efficacy of the experimental intervention under scrutiny, taking into account the numerous facets and intricate variables that contribute to the overall outcome, keeping in mind the statistical significance thresholds and the corresponding probability distributions. Your final judgement should be a synthesis of these insights, crystallized into a ranking that encapsulates the overall potency of the intervention in question.

Label: Assuming Prior Knowledge, Lack of adaptability, Lack of Rating Scale ✔

Explanation:

Assuming Prior Knowledge: The guideline contains terms and concepts such as"quantifiable parameters," "efficacy of the experimental intervention," "statistical significance thresholds," and"probability distributions". These terms assume the annotators process a prior knowledge in statistics or experimental design, which might make them struggle to understand and apply these instructions correctly without specific training or explanation of the professional background.

Lack of adaptability: The guideline requires the annotators to conduct a"comprehensive interrogation of the quantifiable parameters",considering "numerous facets and intricate variables", and also take into account "statistical significance thresholds" and"probability distributions". It's quite strict and complex, leaving little room for adaptability depending on the experimental intervention being evaluated. The rigidness of these instructions can make it challenging for evaluators to apply them across a variety of scenarios or different types of interventions.

Lack of Rating Scale: The guideline suggests that the final judgement should be a"synthesis of these insights, crystallized into a ranking". However,it doesn't provide any clear definition or structure for this ranking system.Without knowing how many points are on the scale or what each point represents,annotators might interpret the ranking system differently, leading to inconsistency in evaluations.

NEGTIVE EXAMPLES

Guideline: This task aims to evaluate the machine translation quality of two different models. You will be given one article and two corresponding translations from the two models in a random order. Your task is to evaluate the quality of the two translation and determine which one you prefer.

Label: Definition Ambiguity, Assuming Prior Knowledge ✗

Explanation:

The guideline does not contain "Definition Ambiguity" error since it has clearly stated the objective of the task. There's no ambiguity about what the annotators are supposed to do, which leaves no room for misunderstanding about how the task is to be carried out. However, it falls under "Lack of Rating Scale" defect because the guideline doesn't provide a specific rating scale or evaluation criteria that the annotators can use to objectively assess the translations. For instance, the guideline could instruct them to rate each translation on a scale of 1-5 for various aspects such as accuracy, fluency, and grammatical correctness, with clear descriptions of what each point on the scale signifies. Besides, the guideline does not has the defect of "Assuming Prior Knowledge". Evaluating the quality of a translation does not inherently require annotators has specialized knowledge about machine translation models, algorithms, or technical jargon. The expectation is that annotators can read and understand both the source and target languages, and are thus capable of judging the quality of the translation.

Table 7: Full instructions given to annotators of the vulnerability detection task in human evaluation guidelines.

[Requirement] = Identify whether the evaluation guideline contains any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".

[Description] = The description of the vulnerabilities is as follows:
Ambiguous Definition: instructions for task definition are unclear, vague, or imprecise that can be interpreted in multiple ways.
Unconscious Bias: instructions unconsciously favors or disadvantages certain results.
Edge Cases: instructions do not specify how to handle edge cases or exceptional situations that don't neatly fit into the usual categories or criteria.
Unclear Rating: instructions that lack standardized criteria for evaluating aspects or definition of each point on a rating scale, resulting in potential inconsistency in ratings.
Prior Knowledge: instructions assume that evaluators have certain background knowledge or familiarity with a specific subject matter, tool, or principle.
Inflexible Instructions: instructions are unnecessarily complex or rigid, making it hard for evaluators to follow and incapable of adjusting to variations in data or task requirements.
Ethical Issues: instructions do not consider potential ethical implications related to the evaluation process, like privacy, cultural sensitivity, accessibility, or the potential misuse of the evaluation results.
Others: covers any defects that do not fall into the above categories.
[Constraints] = Only reply with the name of vulnerabilities or "None". Note that a guideline may contain more than one vulnerability.

[Guideline] = Human Evaluation Guideline: {guideline}

[CoT] = Let's think step by step.

[Examples] =
shot_1 = GUIDELINE: Factual consistency of summarization is defined as the accuracy and faithfulness of a summary in representing the source.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Ambiguous Definition

shot_2 = GUIDELINE: You will be presented with one human-written reference summary first as well as three system summaries generated by trained neural networks respectively. Please evaluate the quality of opinion summaries with respect to the following 4 features: (1) Informativeness; (2) Faithfulness; (3) Consistence; and (4) Grammaticality.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Ambiguous Definition, Unconscious Bias, Unclear Rating

shot_3 = GUIDELINE: You should make pairwise comparison for the summary evaluation to select the best and worst among them.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Edge Cases

shot_4 = GUIDELINE: Evaluate the website's user interface design on a scale of 1 to 10, considering color aesthetics, balance between text and imagery, navigability, font choices, button placements, menu design, adherence to modern design principles, web page loading speed, and responsive design.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Inflexible Instructions

shot_5 = GUIDELINE: Evaluate the use of object-oriented programming (OOP) principles in the code.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Prior Knowledge

shot_6 = GUIDELINE: Evaluate the comments on this public social media post for sentiment analysis.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Ethical Issues

shot_7 = GUIDELINE: Rate the quality of the website.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
ANSWER: Unclear Rating

(Continued from previous page)

[Examples_CoT] =
shot_1 = GUIDELINE: Factual consistency of summarization is defined as the accuracy and faithfulness of a summary in representing the source.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The "source" here usually has scenarios: the input document or the common sense. The task definition of the guideline is unclear and imprecise that can be interpreted in multiple ways. So the answer is "Ambiguous Definition"
ANSWER: Ambiguous Definition

shot_2 = GUIDELINE: You will be presented with one human-written reference summary first as well as three system summaries generated by trained neural networks respectively. Please evaluate the quality of opinion summaries with respect to the following 4 features: (1) Informativeness; (2) Faithfulness; (3) Consistence; and (4) Grammaticality.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: This guideline does not clarify whether the task definition is to evaluate four summaries based on the source review or to evaluate three system summaries based on the reference, causing "Ambiguous Definition". It does not present the reference and generated summaries in a random order, causing participants to Unconscious Bias towards perceiving the reference as higher quality, result in "Unconscious Bias". It does not provide a detailed explanation of 4 rating aspects, leading to multiple interpretations for different evaluators, causing "Unclear rating". So the answer is "Ambiguous Definition", "Unconscious Bias", "Unclear Rating".
ANSWER: Ambiguous Definition, Unconscious Bias, Unclear Rating

shot_3 = GUIDELINE: You should make pairwise comparison for the summary evaluation to select the best and worst among them.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The guideline does not provide guidance on how to handle the edge cases when two summaries have the same quality. So the answer is "Edge Cases"
ANSWER: Edge Cases

shot_4 = GUIDELINE: Evaluate the website's user interface design on a scale of 1 to 10, considering color aesthetics, balance between text and imagery, navigability, font choices, button placements, menu design, adherence to modern design principles, web page loading speed, and responsive design.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The guideline is unnecessarily complex and rigid, making it hard for evaluators to follow and incapable of adjusting to variations in data or task requirements. So the answer is "Inflexible Instructions"
ANSWER: Inflexible Instructions

shot_5 = GUIDELINE: Evaluate the use of object-oriented programming (OOP) principles in the code.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The guideline doesn't provide detail explanation of OOP principles which assumes that evaluators have certain background knowledge or familiarity with a specific subject matter, tool, or principle. So the answer is "Prior Knowledge"
ANSWER: Prior Knowledge

shot_6 = GUIDELINE: Evaluate the comments on this public social media post for sentiment analysis.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The guideline does not specify the collected comments are anonymous or with user consent, disregarding the personal privacy of the commenters. So the answer is "Ethical Issues"
ANSWER: Ethical Issues

(Continued from previous page)

shot_7 = GUIDELINE: Rate the quality of the website.
REQUIREMENT: Identify the guideline contain any of the following vulnerabilities: "Ambiguous Definition", "Unconscious Bias", "Prior Knowledge", "Edge Cases", "Unclear Rating", "Inflexible Instructions", "Ethical Issues", "Others".
Let's think step by step:
REASONING: The guideline does not provide standardized criteria or definition for evaluation aspects or each point on a rating scale, which can lead to inconsistency in ratings. So the answer is "Unclear Rating"
ANSWER: Unclear Rating

(1) Zero-shot Prompt

Basic = [Requirement] + [Constraints] + [Guideline]

VDesc = [Requirement] + [Description] + [Constraints] + [Guideline]

CoT-Basic = [Requirement] + [Constraints] + [Guideline] + [CoT]

CoT-VDesc = [Requirement] + [Description] + [Constraints] + [Guideline] + [CoT]

(2) Few-shot Prompt

Basic = [Requirement] + [Constraints] + [Examples] + [Guideline]

VDesc = [Requirement] + [Description] + [Constraints] + [Examples] + [Guideline]

CoT-Basic = [Requirement] + [Constraints] + [Examples_CoT] + [Guideline]

CoT-VDesc = [Requirement] + [Description] + [Constraints] + [Examples_CoT] + [Guideline]

Figure 2: Full prompts containing Basic, VDesc and CoT used for vulnerability type detection.

TEXT-DAVINCI-003	
Max Tokens	1000
Temperature	0
GPT-3.5-Turbo	
Max Tokens	1000
Temperature	0

Table 8: Hyper-parameters for TEXT-DAVINCI-003 and GPT-3.5-Turbo models.

BERT & XLNet & ALBERT	
Implementation Library	Transformers (Devlin et al., 2019)
Computing Infrastructure	12GB GeForce GTX 1080 Ti
Max Seq Length	256
Optimizer	AdamW
Optimizer Params:	$\beta = (0.9, 0.999), \epsilon = 1e-6$
Learning rate	2e-5
Loss function	BCELoss
Weight Decay	0.01
Maximum Gradient Norm	1.0
Batch size	8
Epochs	6

Table 9: Hyper-parameters for BERT, XLNet and ALBERT models.

Vulnerability Type	Macro-P	Macro-R	Macro-F1	ACC	AUC	Hamming Loss↓
BERT						
Ambiguous Definition	0.43	0.49	0.41	0.51	0.50	0.49
Unconscious Bias	0.53	0.50	0.50	0.89	0.54	0.11
Prior Knowledge	0.43	0.46	0.45	0.77	0.50	0.23
Edge Cases	0.44	0.47	0.45	0.77	0.51	0.23
Unclear Rating	0.65	0.60	0.57	0.69	0.59	0.31
Inflexible Instructions	0.49	0.46	0.47	0.90	0.48	0.10
Ethical Issues	0.47	0.47	0.47	0.86	0.53	0.14
Others	0.49	0.47	0.48	0.91	0.48	0.09
None	0.48	0.49	0.45	0.76	0.51	0.19
XLNET						
Ambiguous Definition	0.39	0.50	0.41	0.52	0.51	0.48
Unconscious Bias	0.48	0.46	0.47	0.89	0.50	0.11
Prior Knowledge	0.44	0.47	0.45	0.78	0.51	0.22
Edge Cases	0.44	0.47	0.45	0.77	0.51	0.23
Unclear Rating	0.55	0.58	0.56	0.65	0.56	0.35
Inflexible Instructions	0.49	0.46	0.47	0.90	0.48	0.10
Ethical Issues	0.49	0.49	0.49	0.87	0.54	0.13
Others	0.49	0.47	0.48	0.91	0.48	0.09
None	0.46	0.47	0.47	0.77	0.55	0.14
ALBERT						
Ambiguous Definition	0.39	0.50	0.41	0.52	0.51	0.48
Unconscious Bias	0.48	0.47	0.47	0.90	0.51	0.10
Prior Knowledge	0.42	0.47	0.44	0.80	0.51	0.20
Edge Cases	0.45	0.48	0.46	0.79	0.53	0.21
Unclear Rating	0.58	0.53	0.49	0.64	0.52	0.36
Inflexible Instructions	0.49	0.48	0.48	0.93	0.49	0.07
Ethical Issues	0.46	0.47	0.47	0.88	0.53	0.12
Others	0.49	0.48	0.48	0.94	0.50	0.06
None	0.47	0.48	0.46	0.77	0.50	0.19

Table 10: Baseline results of each vulnerability type detection.

Vulnerability Type	Zero-shot						Few-shot					
	Macro-P	Macro-R	Macro-F1	ACC	AUC	Hamming Loss↓	Macro-P	Macro-R	Macro-F1	ACC	AUC	Hamming Loss↓
Basic												
Ambiguous Definition	0.42	0.49	0.41	0.66	0.49	0.34	0.61	0.56	0.53	0.70	0.56	0.30
Unconscious Bias	0.50	0.51	0.43	0.65	0.51	0.35	0.52	0.50	0.41	0.60	0.50	0.40
Prior Knowledge	0.44	0.50	0.47	0.88	0.50	0.12	0.48	0.48	0.47	0.82	0.48	0.18
Edge Cases	0.49	0.47	0.47	0.69	0.41	0.31	0.53	0.52	0.52	0.88	0.55	0.12
Unclear Rating	0.43	0.42	0.40	0.41	0.42	0.59	0.51	0.50	0.50	0.53	0.51	0.47
Inflexible Instructions	0.51	0.64	0.40	0.57	0.64	0.43	0.49	0.54	0.47	0.77	0.54	0.23
Ethical Issues	0.49	0.48	0.48	0.87	0.48	0.13	0.57	0.62	0.56	0.85	0.62	0.15
Others	0.48	0.49	0.49	0.95	0.49	0.05	0.55	0.57	0.57	0.97	0.57	0.03
None	0.48	0.51	0.45	0.72	0.43	0.23	0.52	0.54	0.48	0.73	0.55	0.17
Vdesc												
Ambiguous Definition	0.34	0.50	0.40	0.67	0.50	0.33	0.55	0.58	0.48	0.67	0.55	0.33
Unconscious Bias	0.48	0.48	0.48	0.90	0.48	0.07	0.50	0.49	0.46	0.78	0.53	0.22
Prior Knowledge	0.44	0.50	0.47	0.80	0.50	0.12	0.49	0.48	0.48	0.84	0.54	0.16
Edge Cases	0.44	0.47	0.45	0.77	0.54	0.17	0.53	0.56	0.51	0.88	0.61	0.12
Unclear Rating	0.51	0.51	0.48	0.64	0.51	0.36	0.52	0.52	0.43	0.43	0.56	0.57
Inflexible Instructions	0.50	0.50	0.48	0.81	0.50	0.16	0.59	0.57	0.47	0.73	0.65	0.27
Ethical Issues	0.48	0.48	0.48	0.87	0.48	0.08	0.52	0.52	0.51	0.86	0.54	0.14
Others	0.48	0.49	0.49	0.88	0.49	0.04	0.55	0.60	0.57	0.97	0.58	0.03
None	0.48	0.49	0.49	0.75	0.57	0.15	0.53	0.53	0.50	0.74	0.56	0.18
CoT-Basic												
Ambiguous Definition	0.46	0.47	0.46	0.57	0.47	0.43	0.52	0.52	0.51	0.54	0.62	0.46
Unconscious Bias	0.51	0.60	0.29	0.35	0.60	0.65	0.53	0.53	0.53	0.93	0.67	0.07
Prior Knowledge	0.47	0.50	0.48	0.86	0.50	0.14	0.44	0.50	0.47	0.88	0.59	0.12
Edge Cases	0.53	0.57	0.47	0.57	0.57	0.43	0.53	0.54	0.53	0.76	0.65	0.24
Unclear Rating	0.45	0.48	0.29	0.33	0.60	0.67	0.44	0.44	0.37	0.37	0.58	0.63
Inflexible Instructions	0.51	0.56	0.24	0.29	0.56	0.71	0.55	0.53	0.54	0.97	0.64	0.03
Ethical Issues	0.49	0.47	0.44	0.73	0.47	0.27	0.48	0.49	0.49	0.95	0.57	0.05
Others	0.48	0.46	0.47	0.89	0.45	0.11	0.48	0.50	0.49	0.97	0.62	0.03
None	0.50	0.51	0.41	0.61	0.64	0.38	0.47	0.50	0.48	0.75	0.64	0.14
CoT-Vdesc												
Ambiguous Definition	0.47	0.47	0.47	0.55	0.52	0.45	0.60	0.60	0.60	0.67	0.68	0.34
Unconscious Bias	0.54	0.61	0.51	0.81	0.61	0.19	0.54	0.53	0.53	0.89	0.59	0.11
Prior Knowledge	0.46	0.49	0.47	0.84	0.51	0.16	0.68	0.51	0.49	0.86	0.78	0.14
Edge Cases	0.52	0.55	0.50	0.66	0.54	0.34	0.65	0.67	0.64	0.81	0.71	0.19
Unclear Rating	0.49	0.48	0.39	0.39	0.59	0.61	0.65	0.65	0.58	0.58	0.69	0.42
Inflexible Instructions	0.51	0.55	0.23	0.27	0.55	0.73	0.64	0.65	0.65	0.97	0.74	0.07
Ethical Issues	0.50	0.46	0.42	0.63	0.51	0.37	0.56	0.52	0.53	0.93	0.62	0.07
Others	0.48	0.50	0.49	0.96	0.50	0.04	0.48	0.50	0.49	0.96	0.63	0.04
None	0.51	0.50	0.43	0.66	0.68	0.30	0.58	0.57	0.54	0.79	0.69	0.10

Table 11: TEXT-DAVINCI-003 results of each vulnerability type detection.

Vulnerability Type	Zero-shot						Few-shot					
	Macro-P	Macro-R	Macro-F1	ACC	AUC	Hamming Loss↓	Macro-P	Macro-R	Macro-F1	ACC	AUC	Hamming Loss↓
Basic												
Ambiguous Definition	0.47	0.48	0.46	0.55	0.52	0.45	0.57	0.57	0.57	0.60	0.57	0.40
Unconscious Bias	0.50	0.51	0.34	0.44	0.54	0.56	0.52	0.55	0.50	0.77	0.55	0.23
Prior Knowledge	0.49	0.56	0.50	0.75	0.58	0.25	0.51	0.50	0.49	0.80	0.51	0.20
Edge Cases	0.53	0.58	0.47	0.55	0.62	0.45	0.42	0.50	0.46	0.85	0.50	0.15
Unclear Rating	0.50	0.50	0.42	0.47	0.54	0.53	0.29	0.41	0.34	0.51	0.41	0.49
Inflexible Instructions	0.53	0.78	0.41	0.56	0.57	0.44	0.56	0.92	0.58	0.92	0.58	0.08
Ethical Issues	0.48	0.49	0.46	0.78	0.51	0.22	0.70	0.56	0.58	0.92	0.56	0.08
Others	0.48	0.48	0.48	0.93	0.52	0.07	0.54	0.61	0.55	0.90	0.61	0.10
None	0.51	0.53	0.44	0.65	0.62	0.33	0.50	0.57	0.50	0.76	0.45	0.17
Vdesc												
Ambiguous Definition	0.53	0.55	0.50	0.51	0.61	0.49	0.56	0.55	0.54	0.60	0.55	0.40
Unconscious Bias	0.58	0.60	0.49	0.72	0.65	0.28	0.56	0.70	0.55	0.77	0.70	0.23
Prior Knowledge	0.47	0.53	0.44	0.66	0.58	0.34	0.62	0.60	0.61	0.80	0.60	0.20
Edge Cases	0.56	0.64	0.54	0.65	0.69	0.35	0.42	0.49	0.46	0.84	0.49	0.16
Unclear Rating	0.55	0.54	0.47	0.48	0.59	0.52	0.45	0.48	0.43	0.56	0.48	0.44
Inflexible Instructions	0.52	0.71	0.33	0.44	0.61	0.56	0.55	0.91	0.56	0.91	0.48	0.09
Ethical Issues	0.51	0.56	0.45	0.69	0.61	0.31	0.47	0.49	0.48	0.91	0.49	0.09
Others	0.48	0.49	0.48	0.94	0.54	0.06	0.52	0.59	0.52	0.87	0.60	0.13
None	0.54	0.59	0.46	0.65	0.66	0.30	0.50	0.58	0.50	0.75	0.33	0.18
CoT-Basic												
Ambiguous Definition	0.58	0.58	0.58	0.64	0.58	0.36	0.55	0.55	0.54	0.59	0.55	0.41
Unconscious Bias	0.50	0.50	0.45	0.68	0.52	0.32	0.53	0.60	0.51	0.73	0.58	0.27
Prior Knowledge	0.50	0.49	0.47	0.63	0.54	0.37	0.44	0.48	0.46	0.77	0.49	0.23
Edge Cases	0.57	0.61	0.57	0.73	0.69	0.27	0.62	0.69	0.64	0.81	0.69	0.19
Unclear Rating	0.56	0.56	0.56	0.59	0.65	0.41	0.49	0.50	0.45	0.60	0.52	0.40
Inflexible Instructions	0.53	0.81	0.50	0.79	0.66	0.21	0.50	0.50	0.50	0.97	0.63	0.03
Ethical Issues	0.50	0.48	0.47	0.79	0.54	0.21	0.47	0.48	0.48	0.89	0.52	0.11
Others	0.48	0.48	0.48	0.93	0.54	0.07	0.48	0.49	0.49	0.96	0.53	0.04
None	0.51	0.56	0.49	0.71	0.69	0.26	0.49	0.54	0.49	0.74	0.60	0.17
CoT-Vdesc												
Ambiguous Definition	0.59	0.58	0.57	0.66	0.58	0.34	0.60	0.58	0.58	0.64	0.59	0.36
Unconscious Bias	0.50	0.50	0.47	0.76	0.54	0.24	0.50	0.49	0.48	0.79	0.53	0.20
Prior Knowledge	0.50	0.49	0.49	0.69	0.52	0.31	0.60	0.61	0.60	0.79	0.62	0.21
Edge Cases	0.57	0.58	0.58	0.79	0.62	0.21	0.52	0.54	0.52	0.74	0.54	0.26
Unclear Rating	0.53	0.53	0.52	0.58	0.58	0.42	0.60	0.55	0.51	0.65	0.55	0.36
Inflexible Instructions	0.52	0.74	0.50	0.81	0.64	0.19	0.90	0.90	0.90	0.98	0.72	0.02
Ethical Issues	0.51	0.54	0.50	0.81	0.57	0.19	0.58	0.63	0.59	0.87	0.63	0.13
Others	0.48	0.49	0.49	0.95	0.52	0.05	0.48	0.50	0.49	0.97	0.56	0.04
None	0.52	0.55	0.50	0.74	0.63	0.20	0.48	0.58	0.58	0.73	0.52	0.14

Table 12: GPT-3.5-Turbo results of each vulnerability type detection.

Principal for Human Evaluation Guideline
(a) Explicit Task Definition: Provide a concise and precise task description along with the evaluation aspects to preclude misinterpretation or confusion. Use clear language and avoid jargon or technical terms that may not be commonly known to all evaluators. **(b) Unbiased Instructions:** Ensure that all instructions and statements are free from any unconscious bias. Avoid favoring or disadvantaging certain results or approaches. Use neutral language and present the task in a fair and objective manner. **(c) Avoiding Prior Knowledge Assumptions:** Provide sufficient explanations regarding the subject matter, tools, or principles involved. Avoid assuming that evaluators are equipped with specific background knowledge. A good guideline is to make the content easily comprehensible, even for non-expert annotators. **(d) Comprehensive Coverage:** Formulate instructions that cover a broad range of scenarios, incorporating edge cases and exceptional situations that may not fit neatly into predefined categories or criteria. Clearly specify how evaluators should handle such cases and provide guidance on making informed judgments. **(e) Clear Rating Scale and Criteria:** Define a rating scale and provide a clear explanation of the evaluation criteria for each point on the scale. Ensure that evaluators comprehensively grasp the meaning and expectations associated with each rating level so as to promote consistency in ratings and minimize potential confusion. **(f) Simplicity and Flexibility:** Keep the instructions straightforward and easy to understand. Avoid unnecessary complexity or rigid requirements that may make it difficult for evaluators to follow or adapt to variations in data and task requirements. Provide room for reasonable judgment and adaptability within the evaluation process. **(g) Addressing Ethical Issues:** Identify and address potential ethical issues, including guidelines and safeguards, to ensure ethical considerations are upheld throughout the evaluation process. Consider privacy concerns, cultural sensitivities, accessibility requirements, and the potential misuse of evaluation results. **(h) Attach Examples:** It is highly recommended to list positive examples that contain the input presented to the worker or system, and the anticipated results, thus providing crowdworkers with a clearer comprehension of the task. Additionally, listing negative examples can effectively emphasize THINGS TO AVOID by supplying that should not be generated. **(i) Design a good user interface:** A good user interface provides a positive experience for annotators and the design needs to be tailored to the specific needs of the users. For non-expert crowdsourcing, *acquisition interfaces* can be developed to facilitate the execution of crowdsourcing tasks. While for those running the crowdsourcing project, *management interfaces* are required to monitor progress, assess quality, and manage annotators. **(j) Emphasize precautions:** In concluding the guidelines, remind annotators to be careful while carrying out their tasks and outline the annotation requirements, feedback mechanism, and quality assurance processes explicitly so that annotators can manage their time effectively and provide more accurate evaluations. Adherence to these principles facilitates the creation of a human evaluation guideline that is clear, fair, inclusive, and capable of producing reliable and meaningful results.

Table 13: Principal for reliable human evaluation guideline.