CheckEval: Robust Evaluation Framework using Large Language Model via Checklist

Yukyung Lee*
Korea University
Seoul, Republic of Korea
yukyung_lee@korea.ac.kr

Joonghoon Kim*
Korea University
Seoul, Republic of Korea
joonghoon_kim@korea.ac.kr

Jaehee Kim
Korea University
Seoul, Republic of Korea
jaehee_kim@korea.ac.kr

Hyowon Cho
KAIST
Seoul, Republic of Korea
hyyoka@kaist.ac.kr

Pilsung Kang
Korea University
Seoul, Republic of Korea
pilsung_kang@korea.ac.kr

ABSTRACT

We introduce CheckEval, a novel evaluation framework using Large Language Models, addressing the challenges of ambiguity and inconsistency in current evaluation methods. CheckEval addresses these challenges by dividing evaluation criteria into detailed sub-aspects and constructing a checklist of Boolean questions for each, simplifying the evaluation. This approach not only renders the process more interpretable but also significantly enhances the robustness and reliability of results by focusing on specific evaluation dimensions. Validated through a focused case study using the SummEval benchmark, CheckEval indicates a strong correlation with human judgments. Furthermore, it demonstrates a highly consistent Inter-Annotator Agreement. These findings highlight the effectiveness of CheckEval for objective, flexible, and precise evaluations. By offering a customizable and interactive framework, CheckEval sets a new standard for the use of LLMs in evaluation, responding to the evolving needs of the field and establishing a clear method for future LLM-based evaluation.

CCS CONCEPTS

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**.

KEYWORDS

Large Language Model, GPT, Open-ended Task, LLM-based Evaluation

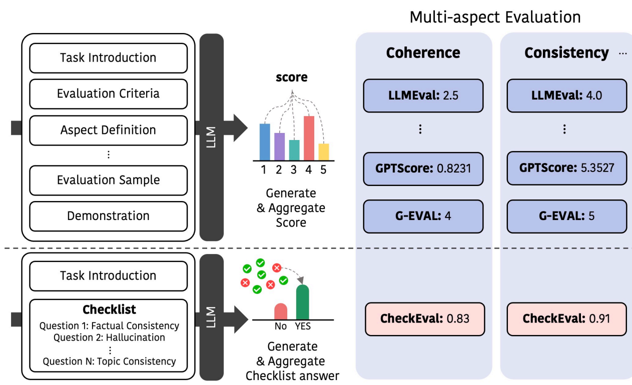

Figure 1: Illustration of the various LLM-based evaluators and CheckEval. Unlike most LLM-based evaluators, CheckEval generates responses to the checklist and aggregates these responses to the final quality score.

1 INTRODUCTION

The advancement of Large Language Models (LLMs) has expanded the capabilities of AI models across various domains, offering new tools and opportunities to researchers and developers alike [3, 7, 34]. Notably, models like GPT-4 have demonstrated performance beyond the previous limitations in various applications such as conversational agents, automated content creation, and arithmetic reasoning [1]. Furthermore, the use of LLMs as evaluators for generated text has emerged as a novel approach [5, 13, 28, 37, 45]. This approach offers an efficient mechanism for assessing the quality of open-ended generation and comparing model performances, primarily aiming to reduce the time and computational costs associated with traditional human evaluation methods.

Previous studies using LLMs as evaluators typically involve providing definitions and instructions for specific tasks, which then guide the models in conducting assessments. For instance, one approach applies Chain-of-Thought (CoT) prompting [38] to LLMs, utilizing a form-filling paradigm to generate explicit scores for the generated text [28]. Another method leverages probabilities to predict evaluation scores for a final assessment [13].

However, assessment outputs from LLM-based evaluators often have unclear criteria, making it difficult to distinguish the quality of text as models become more advanced, such as moving from GPT-3 to GPT-4 [6]. In particular, the Likert scale evaluation system (e.g., 1-5) [8, 17], utilized in both LLM and human assessments, struggles to consistently evaluate aspects such as 'fluency' and 'coherence' [23, 27]. Furthermore, the difficulty in reproducing identical outputs with the Likert scale evaluation system highlights the need for reconsideration and improvement [19].

Motivated by the imperative to address the challenges of unclear evaluation criteria and complexities, we introduce **CheckEval**, a novel evaluation framework. Building on the insights from recent advances in fine-grained analysis regarding the evaluation process by Liu et al. [27] and Min et al. [30], CheckEval decomposes

*Both authors contributed equally to this research.

HEAL@CHI'24, May 12–05, 2024, Honolulu, Hawaii, USA
2024.

evaluation criteria into more detailed sub-aspects and develops a checklist for each dimension. This approach does not generate scores directly during the evaluation process; instead, it breaks down the evaluation into discrete, Boolean questions, prompting LLMs to respond to the checklist. This decomposition simplifies the evaluation process and enhances the explainability of assessments. It also significantly improves agreement among evaluators, ensuring evaluation is consistent and reliable across various models and evaluators. This enhanced agreement provides a dependable basis for objectively comparing model performance. Furthermore, the adaptability of CheckEval to the evolving needs of instruction-based LLMs for new task evaluations is significant. CheckEval rises to this challenge by offering evaluations that are both customizable and interactive, specifically designed to meet the varied needs of different applications. This framework enables the detailed definition and assessment of specific evaluation criteria, thereby aiming to enhance the flexibility and accuracy of evaluations across diverse domains.

To validate the effectiveness of CheckEval, we conducted a case study utilizing the widely adopted SummEval benchmark [10]. This case study entailed evaluating a subset of SummEval to explore the capabilities of CheckEval in depth. The analysis focused on the correlation between LLM evaluations and human judgments, as well as measuring the Inter-Annotator Agreement (IAA) [2] across different evaluation models within the same framework. The findings from these preliminary results indicate that CheckEval significantly clarifies the evaluation process and enhances consistency among different evaluators.

2 RELATED WORK

2.1 LLM-based Evaluators

Recent studies have demonstrated that traditional metrics such as ROUGE [25] and BLEU [32] are incapable of accurately evaluating open-ended generation tasks due to their inherent dependence on reference text [6, 15, 16, 33, 40]. Recent advancements in LLMs have led to the emergence of research using LLMs as evaluators, demonstrating their potential in overcoming these limitations [20, 21, 26, 29]. Notably, approaches using powerful LLMs as evaluators, such as GPT-4, have shown remarkable performance [13, 28]. However, current LLM-based evaluators exhibit shortcomings in terms of robustness, as their performances are very sensitive to the prompts, resulting in instability of the evaluation process. Furthermore, when LLM-based evaluators are provided with the Likert scale evaluation system, the interpretation of the difference between scores may not be clear, potentially compromising the reliability of their evaluation outputs [23]. Recent studies have attempted to mitigate these challenges by focusing on the generation of explanations for evaluation outputs [6]. However, generating explanations does not intrinsically enhance robustness or reliability due to issues such as hallucination [41]. In this study, CheckEval adopts a binary evaluation system characterized by clear criteria to address this ambiguity. Through this approach, we aim to enhance the robustness and reliability of the LLM-based evaluator.

2.2 Decomposition Strategy

In various tasks, several approaches decompose complex information into minimal units to simplify the difficulty of the task [4, 11, 18, 22, 31, 35, 39, 43]. Specifically, Liu et al. [27] and Min et al. [30] have demonstrated that decomposing complex content to atomic units reduces the subjectivity of the judgment of factual consistency. Atomic fact units are conceptualized in Liu et al. [27] as "elementary information units in the reference summaries which no longer need to split." They are described in Min et al. [30] as "a short sentence conveying one piece of information, similar to summarization content units [31]".

In line with these studies, CheckEval aims to reduce task complexity to improve the objectivity and robustness of evaluations. However, unlike previous studies, CheckEval does not decompose content (summary in summarization task) but rather evaluation criteria, which simplifies the evaluation process. Furthermore, CheckEval is not constrained to any specific aspect or task; rather, it functions as a flexible framework that is applicable across a broad range of evaluative dimensions.

3 DESIGN OF CHECKEVAL

CheckEval introduces a structured framework for the evaluation of generated text, as elaborated in Figure 2. This framework comprises three distinct stages: 1) Aspect Selection, 2) Checklist Generation, and 3) Checklist-based Evaluation. The core idea of CheckEval involves constructing a checklist by decomposing aspect into subcomponents using a question format. The questions within the checklist are formatted in a Boolean QA style, allowing for binary responses (Yes/No). This format improves the precision and clarity of evaluation, facilitating a more straightforward interpretation and offering an advantage over traditional 1-5 scale ratings.

In the Aspect Selection stage, humans select specific aspects and define key components within each aspect to align assessments with the objectives of the task. This stage results in the creation of a draft checklist that provides explicit guidelines and rubrics for evaluating designated tasks. In the Checklist Generation stage, based on the selected aspects and key components, a checklist is constructed through three steps: (a) Key Questions Writing, (b) Questions Augmentation, and (c) Question Filtering. This stage involves generating various questions for the checklist with Boolean Question Answering (QA) format and filtering them to finalize the checklist. In the Checklist-based Evaluation stage, CheckEval leverages the curated checklist to evaluate the quality of the generated text. LLMs generate responses to the questions on the checklist. Subsequently, these responses are aggregated to compute a final score.

3.1 Aspect Selection

As shown in Figure 2, the first step in developing the checklist involves selecting aspects that align with the specific objectives and intended outcomes of the tasks. In this step, humans can either define custom aspects or select from widely used aspects (e.g., consistency, fluency). The next step is to define key components for each aspect. These key components represent sub-concepts that capture the meaning of the aspect and are identified as essential elements for it. For example, in summarization tasks, grammatical

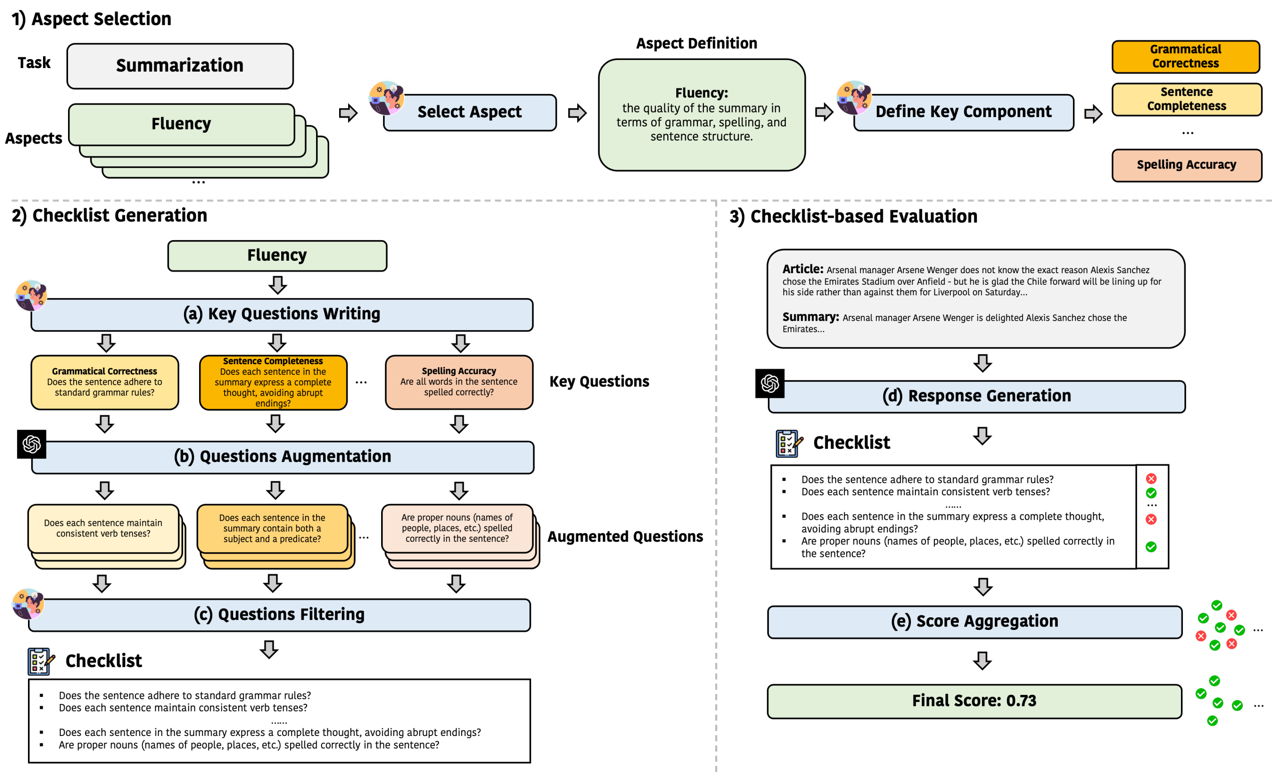

Figure 2: Overall process of CheckEval. CheckEval consists of three stages. 1) In the Aspect Selection stage, humans select specific aspects and define key components. 2) In the Checklist Generation stage, 3) In the Checklist-based Evaluation stage

correctness, sentence completeness, and spelling accuracy are regarded as key components of the 'fluency' aspect. By specifying key components that capture the core of each aspect, this process provides evaluators with clear and explicit criteria for assessment.

3.2 Checklist Generation

The creation of the checklist consists of three steps: (a) Key Questions Writing, (b) Questions Augmentation, and (c) Questions Filtering.

(a) Key Questions Writing In this step, a single question is formulated for each key component. The key questions are structured in Boolean QA format, allowing only for 'Yes' or 'No' responses. This approach is designed to reduce the complexity involved in making judgments [23, 27]. For example, evaluators can more easily respond to "Does the sentence adhere to standard grammar rules?" with 'Yes' or 'No', compared to "How does the sentence adhere to or deviate from standard grammar rules?" with 1-5 Likert scale. In this work, the authors carefully construct each key question to cover specific aspects, ensuring minimal overlap in semantics.

(b) Questions Augmentation In this step, each key question undergoes expansion via LLM. The objective is to further dissect the key components into more granular components, thereby generating an expanded set of questions rooted in the original key questions. Augmentation of questions is guided by two primary objectives: 1) to encompass the underlying meaning of the aspect and its key component, and 2) to be more specific than the original key questions. For example, from the key question like For example, a key question, "Are all words in the sentence spelled correctly?" can be augmented with more specific questions, "Are proper nouns(names of people, places, etc.) spelled correctly in the sentence?". Similar to key questions, augmented questions are also presented in a Boolean QA format. [1] [2]

(c) Questions Filtering To ensure a high-quality checklist, we apply a filtering process for both key and augmented questions. This process reviews the clarity of each question, identifies redundancies, and ensures alignment with the intended evaluation aspect. We only retain the questions that directly relate to the evaluative objectives. This selection process leads to a highly refined and targeted final checklist. [3] [4]

3.3 Checklist-based Evaluation

In this stage, as illustrated in Figure 2 (d) and (e), first utilizes LLM to generate responses directly to the individual questions within the checklist, then aggregates the responses to calculate the

[1]Each question is designed so that a 'Yes' response indicates adherence to desired criteria, as illustrated by the question, 'Does the summary avoid introducing information not present in the article?

[2]The process of question augmentation is executed utilizing GPT-4, employing prompts that integrate aspects definition, key components, and key questions.

[3]The filtering process typically retains 3-5 questions per key component, resulting in an average of 4 questions per component.

[4]The filtering process is carried out by the authors.

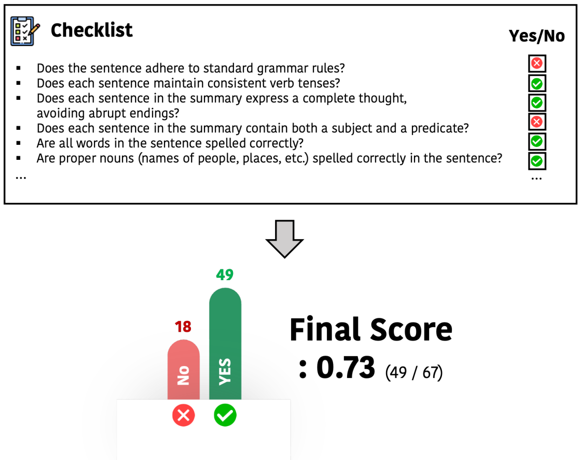

Figure 3: Illustration of Score Aggregation stage. The proportion of 'Yes' answers to the total number of questions in the checklist is used as the final score.

score. In this stage, as shown in Figure 2(d), CheckEval employs LLM to generate responses directly to each question within the checklist. Unlike traditional automatic metrics, which typically yield a single quality score for each generated text, the CheckEval produces multiple responses for each generated text. Therefore, to convert the multiple responses for each generated text into a single score, score aggregation is conducted. CheckEval treats the proportion of positive responses ('Yes' answers) to the total number of questions used as the final score (see Figure 3). This approach rests on the assumption that the ratio of positive responses can serve as a proxy for overall text quality, given that each question addresses a subcomponent of an evaluation criteria. Through this process, CheckEval offers inherent explainability by characteristics of its checklist-based assessment, even without the LLM explicitly generating rationales for judgment.

4 CASE STUDY

4.1 Evaluation Setting

4.1.1 Dataset. We evaluate our CheckEval on SummEval [10], one of the most widely used benchmarks in the summarization task. SummEval provides source texts, reference texts, and model-generated texts from various summarization models, along with evaluation aspects and human scores. SummEval consists of human annotations on the quality of summaries across four aspects: coherence, consistency, fluency, and relevance. To effectively assess the feasibility of our framework, we sample 10% of the entire data. This sampling process is performed to reflect the distribution of human annotations across each aspect uniformly.

4.1.2 Measuring Performance. We employ sample-level correlation to measure the performance of automatic metrics. Sample-level correlation is calculated for each sample individually based on outputs from multiple systems and then averaged across all samples [13]. For each source text $s_i, i \in \{1, 2, \ldots, n\}$ (e.g., documents in summarization task), there exist J system outputs $h_{i,j}$ (e.g., summaries in summarization task), where $j \in \{1, 2, \ldots, J\}$. f_{auto} is scoring function of automatic metric (e.g., ROUGE [25]), and f_{human} is the gold human scoring function. The sample-level correlation C for each aspect is defined as follows:

$$C^{\text{sample}}_{f_{\text{auto}}, f_{\text{human}}} = \frac{1}{n} \sum_{i=1}^{n} \Big(g \big(\big[f_{\text{auto}}(h_{i,1}), \ldots, f_{\text{auto}}(h_{i,j}) \big], \big[f_{\text{human}}(h_{i,1}), \ldots, f_{\text{human}}(h_{i,j}) \big] \big) \Big),$$

where g can be either Spearman or Kendall's tau correlation.

4.1.3 Baselines. We compare our CheckEval with the following baselines: (1) BERTScore [42] calculates text similarity by contextual embeddings of BERT [9]. (2) MoverScore [44] enhances BERTScore by incorporating soft alignments and novel aggregation approaches for a more effective similarity measure. (3) BARTScore [42] serves as an unified evaluator, leveraging the pretrained BART [24] which uses average likelihood of the model output. (4) UniEval [46] is a unified multi-dimensional evaluator that can assess various aspects of text generation tasks. (5) G-Eval [28] utilizes LLMs like GPT-4 for text quality evaluation. It employs a chain-of-thoughts approach and a form-filling paradigm to assess the quality of texts. [5]

4.1.4 CheckEval. Checklists are constructed for each of the four aspects used by SummEval. Subsequent to the process of question filtering, the finalized checklist comprises varying numbers of questions for each aspect: 21 for coherence, 13 for consistency, 15 for fluency, and 18 for relevance. The prompt used for question augmentation is detailed in Appendix Table 3. In the Checklist-based Evaluation stage, LLM is required to generate responses to approximately 4-5 questions at once. The prompt used for response generation is detailed in Appendix Table 4. In the experiment, we use models such as GPT-3.5-turbo (`gpt-3.5-turbo`), GPT-4 (`gpt-4`), and GPT-4-turbo (`gpt-4-1106-preview`).

4.2 Case Study 1: Correlation Analysis

To evaluate the performance of CheckEval, we conduct a comparative analysis against baseline evaluation methods. Following previous works [13, 28, 46], we examine the correlation between CheckEval and human annotation scores using Spearman and Kendall-Tau coefficients to ensure the reliability of evaluations.

As shown in Table 1, CheckEval outperforms traditional automatic metrics like Rouge, BLEU, METEOR, BERTScore, and BARTScore in terms of correlation coefficients. It indicates that CheckEval could be a more human-like evaluation method compared to traditional automatic metrics. Compared to UniEval, CheckEval consistently shows better performance. When matched against G-Eval, CheckEval is comparable or superior. Notably, CheckEval with GPT-4 achieved the highest Kendall Tau correlation among consistency, fluency, and relevance of all metrics, highlighting its exceptional average performance. This performance demonstrates the detailed approach of CheckEval, which breaks down criteria for thorough analysis, leading to strong correlations with human evaluations. The coherence correlation observed in CheckEval is lower

[5] For G-Eval, we utilized prediction results made publicly available in the SummEval official repository (https://github.com/Yale-LILY/SummEval)

Evaluation Methods	Coherence		Consistency		Fluency		Relevance		Average	
	spearman	kendall	spearman	kendall	spearman	kendall	spearman	kendall	spearman	kendall
Non-LLM based metric										
ROUGE-L	0.1722	0.1217	0.3170	0.2354	0.2402	0.1666	0.4204	0.3011	0.2599	0.1871
BLEU	0.0277	0.0192	0.0362	0.0297	-0.0786	-0.0566	0.4227	0.3081	0.0225	0.0169
METEOR	0.0085	0.0078	0.1005	0.0735	0.0106	0.0079	0.2746	0.1820	0.0549	0.0397
BERTScore	-0.2194	-0.1507	-0.1086	-0.0762	-0.1562	-0.1128	0.6423	0.4761	-0.1882	-0.1336
MOVERSscore	0.3398	0.2522	0.3844	0.2767	0.3447	0.2467	-0.4336	-0.3151	0.3687	0.2680
BARTScore	0.4745	0.3426	0.4565	0.3414	0.3952	0.2849	0.6023	0.4551	0.4513	0.3301
UniEval	0.5669	0.4113	0.6129	0.4762	0.6050	0.4470	**0.6399**	<u>0.4761</u>	0.5551	0.4122
LLM based metric										
G-Eval (GPT4)	**0.6193**	**0.4703**	<u>0.6642</u>	0.5171	<u>0.6288</u>	<u>0.4822</u>	0.6166	0.4637	**0.6322**	0.4833
CheckEval (GPT3.5-turbo)	0.4449	0.3474	0.5998	0.5146	0.3927	0.3265	0.3045	0.2520	0.4355	0.3601
CheckEval (GPT4)	0.5731	0.4279	**0.7062**	**0.6106**	**0.6320**	**0.4931**	0.5698	0.4384	<u>0.6203</u>	**0.4925**
CheckEval (GPT4-turbo)	<u>0.5942</u>	<u>0.4437</u>	0.6505	<u>0.5509</u>	0.6127	0.4722	<u>0.6184</u>	**0.4870**	0.6189	<u>0.4884</u>

Table 1: Sample-level Spearman (ρ) and Kendall tau (τ) correlations of different aspects on SummEval benchmark. The best overall results are highlighted in bold. The second best results are <u>underlined</u>.

Fleiss' Kappa	Coherence	Consistency	Fluency	Relevance
2 Models				
GPT-(3.5-turbo, 4)	0.4206	0.4085	0.0806	0.3160
GPT-(3.5-turbo, 4-turbo)	0.4530	0.4098	0.0827	0.3167
GPT-(4, 4-turbo)	**0.7199**	**0.7181**	**0.4788**	**0.6968**
3 Models				
GPT-(3.5-turbo, 4, 4-turbo)	0.5289	0.5127	0.2612	0.4644

Table 2: Agreement measures (Fleiss' Kappa) on different models. We consider GPT-3.5-turbo, GPT-4, and GPT-4-turbo as evaluators and measure agreement level. The best overall results are highlighted in bold.

than G-Eval; however, this minor degradation does not significantly affect the average correlation of CheckEval. These results demonstrate the broad capability of CheckEval for conducting accurate evaluations across varied quality aspects.

4.3 Case Study 2: Robustness Analysis

To assess the robustness of CheckEval, this study utilizes Fleiss' kappa [12], a statistical method commonly applied to measure the level of agreement among multiple evaluators. In this experiment, while keeping the input prompts or instructions constant, we change only the models used for evaluation, treating each model as a 'rater'. This approach aims to quantitatively assess the robustness of CheckEval against changes in evaluation models through the consistency of scores across different models.

Fleiss' kappa values range from -1 to 1, with values closer to 1 indicating perfect agreement among raters, 0 indicating agreement by chance, and negative values indicating less agreement than would be expected by chance. We examine the IAA between two models (GPT-3.5-turbo and GPT-4) and among three models (GPT-3.5-turbo, GPT-4, GPT-4-turbo) to observe patterns of agreement across models.

The analysis shows that the agreement between GPT-3.5-turbo and other models yields relatively lower values due to performance differences, with evaluation scores of GPT-3.5-turbo significantly differing from those of GPT-4 and GPT-4-turbo, a finding corroborated by correlation analysis results. Conversely, a higher Fleiss' kappa value is observed between GPT-4 and GPT-4-turbo, indicating a very high level of score agreement between these models. This high degree of agreement demonstrates that CheckEval exhibits a high level of robustness, capable of providing consistent evaluation outcomes despite changes in the evaluation models.

5 FUTURE WORK

This ongoing research aims to demonstrate the generalization capability of the proposed framework and enhance research efficiency through the automation of the evaluation process.

Extending Task Coverage Future work will expand CheckEval to cover more datasets and evaluation tasks, aiming to explore its versatility. The plan includes assessing a range of NLP tasks and datasets, such as Topical Chat [14] and QAGS [36], to test its effectiveness across different settings.

Score Aggregation This study employed a score aggregation method to convert individual checklist outputs into a single score for measuring text quality. Currently, responses to each question are assigned values of 1 for 'Yes' and 0 for 'No', with the score determined by averaging the proportion of positive responses. We assume that the number of positive answers can represent the text quality, considering each question as an element of the aspect of quality being assessed. Future research, however, plans to explore more effective score aggregation methods to enhance the precision and reliability of evaluations.

Question Filtering In the current study, data filtering was conducted manually by the authors, indicating room for improvement. Future efforts will focus on developing automated data filtering techniques to minimize the need for human effort.

6 CONCLUSION

This paper introduced CheckEval, a novel evaluation framework for LLMs. CheckEval leverages a structured evaluation checklist to analyze NLG system outputs, aiming to enhance the precision and clarity of evaluation. Initial validation on a subset of the SummEval dataset demonstrated the robust performance of CheckEval, indicating its ability to offer detailed and interpretable evaluation results. Notably, CheckEval achieved a strong correlation with human evaluations and consistent agreement across different models, emphasizing its potential for reliable assessments.

Future efforts will seek to broaden the scope of validation by including the full SummEval dataset and extending it to other open-ended text generation tasks. Such expansions aim to thoroughly evaluate the adaptability and effectiveness of CheckEval across various tasks. Positioned at the forefront of LLM evaluation innovation, CheckEval is anticipated to evolve as a significant methodology in the advancement of LLM-based evaluation techniques, with its value expected to expand through future research and practical application.

ACKNOWLEDGMENTS

We are particularly grateful to the Unsupervised Learning team at Naver: Jungsoo Park, Soonwon Ka, Bokyung Son, and Jaewook Kang, whose insights were invaluable to our research. We also thank Keonwoo Kim for the discussion and feedback on the paper.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Ron Artstein. 2017. Inter-annotator agreement. *Handbook of linguistic annotation* (2017), 297–313.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. *arXiv preprint arXiv:2205.06938* (2022).

[5] Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the Use of Large Language Models for Reference-Free Text Quality Evaluation: An Empirical Study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (Eds.). Association for Computational Linguistics, Nusa Dua, Bali, 361–374. https://aclanthology.org/2023.findings-ijcnlp.32

[6] Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations?. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 15607–15631. https://doi.org/10.18653/v1/2023.acl-long.870

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.

[8] LA Clark. [n. d.]. 8c Watson, D.(1995). Constructing validity: Basic issues in objective scale development. *Psychological Assessment* 7, 3 ([n. d.]), 309–319.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

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

[11] Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. *arXiv preprint arXiv:2011.05448* (2020).

[12] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

[13] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166* (2023).

[14] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*. 1891–1895. https://doi.org/10.21437/Interspeech.2019-3079

[15] Jing Gu, Qingyang Wu, and Zhou Yu. 2021. Perception score: A learned metric for open-ended text generation evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12902–12910.

[16] Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2021. Openmeva: A benchmark for evaluating open-ended story generation metrics. *arXiv preprint arXiv:2105.08920* (2021).

[17] Timothy R Hinkin. 1998. A brief tutorial on the development of measures for use in survey questionnaires. *Organizational research methods* 1, 1 (1998), 104–121.

[18] Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *arXiv preprint arXiv:2303.01432* (2023).

[19] Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. GENIE: Toward reproducible and standardized human evaluation for text generation. *arXiv preprint arXiv:2101.06561* (2021).

[20] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491* (2023).

[21] Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520* (2023).

[22] Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. *arXiv preprint arXiv:2301.13298* (2023).

[23] Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9662–9676. https://doi.org/10.18653/v1/2023.emnlp-main.600

[24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[25] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[26] Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. 2023. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects. *arXiv preprint arXiv:2311.08788* (2023).

[27] Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the Gold Standard: Grounding Summarization Evaluation with Robust Human Evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 4140–4170. https://doi.org/10.18653/v1/2023.acl-long.228

[28] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[29] Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. Calibrating llm-based evaluator. *arXiv preprint arXiv:2309.13308* (2023).

[30] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12076–12100. https://doi.org/10.18653/v1/2023.emnlp-main.741

[31] Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*. 145–152.

[32] 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*. 311–318.

[33] Kamila Polišenská, Shula Chiat, Jakub Szewczyk, and Katherine E Twomey. 2021. Effects of semantic plausibility, syntactic complexity and n-gram frequency on children's sentence repetition. *Journal of Child Language* 48, 2 (2021), 261–284.

[34] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[35] Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. *arXiv preprint arXiv:1904.05929* (2019).

[36] Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). https://doi.org/10.18653/v1/2020.acl-main.450

[37] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a Good NLG Evaluator? A Preliminary Study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini (Eds.). Association for Computational Linguistics, Singapore, 1–11. https://doi.org/10.18653/v1/2023.newsum-1.1

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[39] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. *arXiv preprint arXiv:2203.12990* (2022).

[40] Haiyan Wu, Zhiqiang Zhang, and Qingfeng Wu. 2021. Exploring syntactic and semantic features for authorship attribution. *Applied Soft Computing* 111 (2021), 107815.

[41] Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback. *arXiv preprint arXiv:2305.14282* (2023).

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

[43] Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. *arXiv preprint arXiv:2109.11503* (2021).

[44] 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* (2019).

[45] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id=uccHPGDlao

[46] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197* (2022).

A PROMPTS

Prompt
In this task, you need to create a question to evaluate the {aspect} of the summary of the original document. The definition of {aspect} and the questions corresponding to the key component of {aspect} are provided below. Use them to generate sub-questions for each key question. Each sub-question must satisfy the following conditions: 1. Each question must be answerable with 'Yes' or 'No'. 2. Each question must contain concepts from the key component. 3. Each question should minimize the subjectivity of the rater's judgment. 4. The semantic redundancy between sub-questions should be minimized. 5. Formulate questions so that a 'Yes' answer is a positive answer. # Definition {aspect} - {definition} # Key component and corresponding question - {key component}: {key question} Sub-questions:

Table 3: Questions Augmentation Prompt for Checklist Generation Stage.

Prompt
In this task, you will be provided with a news article and a summary. Your task is to answer 'Yes' or 'No' to the questions related to the {aspect}. Do not generate any explanations without answer to the questions. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed. Evaluation Criteria: {aspect} - {definition} Evaluation Steps: 1. Analyze the summary to evaluate {aspect}. 2. Respond to each of the following questions with either 'Yes' or 'No' to evaluate the {aspect}. 3. Please answer 'Yes' or 'No'. No need to any explain. Article: {source} Summary: {summary} Questions: - {question} - {question} ... Your Answers:

Table 4: Answer Generation Prompt for Checklist-based Evaluation Stage.