生成系タスクの自動評価において チェックリストの使用は有効なのか？

古橋萌々香 [1,2]　中山功太 [2]　児玉貴志 [2]　菅原朔 [3,2]
[1] 東北大学　[2]NII LLMC　[3]NII
furuhashi.momoka.p4@dc.tohoku.ac.jp　{nakayama,tkodama,saku}@nii.ac.jp

概要

生成系タスクにおける大規模言語モデルを用いた自動評価では，評価基準の曖昧さが課題とされている．これに対し，チェックリストにより評価基準を細分化する方法が注目されているが，作成方法の検討はまだ十分でない．本研究では 6 つの生成手法でチェックリストを作成し，それに基づいて回答を評価，その有効性を 3 種の評価モデルで検証した．その結果，チェックリスト未使用時と比較して一致率が向上したケースは 22.6%に留まり，チェックリストの有効性は限定的であった．一方，項目数制限や付加情報の活用が有効であり，小規模評価モデルでも適切なチェックリストを用いることで大規模モデルと同等の評価を行える可能性が示唆された．

1　はじめに

大規模言語モデル（LLM）は，多肢選択問題のような分類タスクからコーディングのような生成タスクまで，幅広いタスクに対応できる汎用性を備えている．近年ではそうした LLM が生成する応答の質を評価することが，LLM の開発プロセスにおいて重要な要素となりつつある．分類タスクの評価は，回答と正解の一致度を計算することで比較的容易に行える．一方で，生成タスクの評価は流暢性や一貫性などの評価指標としてペアワイズ評価や 5 段階評価が行われることが多いが，これらの評価基準は曖昧であり，アノテータ間の相関ですらあまり高くない．ましてや，近年注目されている LLM を用いた自動評価においては，自動評価と人手評価の相関が低い点が課題となっている．

この課題に対し，曖昧な評価基準を細分化した「チェックリスト」を評価に導入する研究がある [1, 2, 3]．ただ，こうした研究のチェックリストの多くはどの要素を含めるべきか（または含めないべきか）について十分な検討がされておらず，「どのような特徴を含むチェックリストが自動評価に効果的であるか」は明らかになっていない．

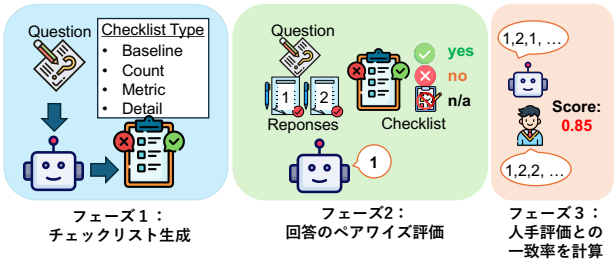

図 1: チェックリスト生成と生成系タスクの評価の流れ．チェックリストでは，チェックリスト項目数の制限 (Count)，評価指標の導入 (Metric) およびチェックリスト内容の詳細性の調整 (Detail) の 3 種の生成手法を検討する．回答の評価では，「yes」「no」「n/a」の 3 つのラベルを使用する．

本研究ではチェックリストの作成方法に注目し，自動評価における効果的なチェックリストの特徴を検証する．本研究の流れを図 1 に示す．まず既存の評価ベンチマークの入力質問をもとに評価の際に確認すべきチェックリストを LLM （本稿では GPT-4o [4] と Claude 3.5 Sonnet [5]）に生成させる（フェーズ 1）．この際，チェックリスト項目数の制限，評価指標の導入，チェックリスト内容の詳細性の調整の 3 つの生成手法でそれぞれチェックリストを生成させ比較検討する．また，生成させたチェックリストをもとに，「yes」，「no」，「n/a」の 3 種類のラベルを使用して回答をペアワイズで自動評価する（フェーズ 2）．この評価には GPT-4o, Qwen2.5-32B-Instruct [6], Qwen2.5-7B-Instruct [6] の 3 種類の言語モデルを使用し，評価モデルのパラメータ数の影響を調査する．最後に自動評価結果と，評価ベンチマークに付随している人手評価結果から一致率を計算し，チェックリスト未使用の場合の自動評価結果と比較する（フェーズ 3）．

実験結果より，一致率が向上したケースは全体の22.6%であり，チェックリストが有効な場合は限定的であることが明らかになった．一方で，チェックリスト項目数制限やチェックリスト内容の詳細性の調整の 2 つの生成手法は 3 種の評価モデル全てで有効である傾向が確認された．また，評価モデルが小規模であっても，適切に生成されたチェックリストを使用することで，高性能な LLM と同等以上の評価結果を得られる可能性があることが示唆された．

2 チェックリスト生成手法

本研究では LLM を活用し，自動評価に対応したチェックリストの生成手法の比較を行う．予備実験の結果よりチェックリスト生成には次の 3 つの課題が存在することがわかった：(1) 生成されるチェックリストの数が多すぎる．(2) 回答の評価に直接的に必要のないチェックリストが生成される．(3) チェックリストの内容が表面的なものに留まり，具体性に欠く場合がある．

これらの課題に対処するため，本研究では生成系タスクを評価する際に使用するチェックリストを (1) **チェックリスト項目数の制限**，(2) **評価指標の導入**，(3) **チェックリスト内容の詳細性の調整**の 3 種類の手法で生成した．以下にベースラインも含めた各手法の詳細を記す．

ベースライン (Baseline) 本研究では，チェックリストの作成手法を比較することを目的としている．比較の基準として，Cook ら [7] が提案した LLM を用いたチェックリスト作成手法をベースラインとして採用した．Cook らのプロンプトには「各質問に対するチェックリストの最小項目数を 2 個，最大項目数を 8 個とする」という項目数制限が含まれていたが，本研究ではこの制限を削除し，チェックリストの生成を行った．

チェックリスト項目数の制限 (Count) InfoBench [2] や WildBench [3] は回答に対するチェックリストの項目数の制限を設けていない．しかし，過剰なチェックリストは評価者に対して過度な負担を強いる可能性があり，細分化が進みすぎることで混乱を招く恐れがある．これらの問題を避けるため，本研究ではチェックリストの最大項目数を 3 個 (count_3)，5 個 (count_5)，10 個 (count_10) の 3 つのパターンで生成する．

評価指標の導入 (Metric) 従来の生成系タスクの評価では多様な評価指標が使用されてきた一方で，これらの評価指標をもとにしたチェックリストの生成は十分に検討されていない．そこで本研究では，様々な生成系タスクの評価で使用されている指標の中から流暢性，関連性，一貫性，明瞭性，正確性，網羅性，詳細性，独自性，有用性，共感性，安全性，中立性，適切性の 13 種類の指標を選び出し，各入力質問ごとにこれらの指標の中から必要な指標を使用してチェックリストを生成する．

チェックリスト内容の詳細性の調整 (Detail) 先行研究のチェックリストでは，回答を表面的に評価するもの (例：この回答は正確であるか) とその内容に踏み込んだもの (例：富士山の標高は 3,776m であるか) の 2 つのタイプが存在していた．そこで本研究では，質問に対して，どのような回答が来てもチェックリストとして使用できる必要条件 (detail) のみを含む項目と，内容に踏み込んだ付加条件 (detail++) も含む項目の 2 種類のチェックリストを生成する．

上述の 4 手法に共通して，各チェックリスト項目は「yes」または「no」で回答可能な形式とする．チェックリストの生成には GPT-4o と Claude 3.5 Sonnet を使用する．出力結果が指定するフォーマットに該当しない場合は最大 3 回までチェックリストの生成を実施する．

3 評価実験

3.1 実験設定

本実験では，LLMBar データセット [8] に含まれる Adversarial (GPTInst, GPTOut, Manual, Neighbor) と Natural を使用した．Adversarial は評価者である LLM を誤解させる傾向のある質問で設計されており，Natural は既存の人間の嗜好に基づくデータセットから質問を収集し，各質問に対して客観的な回答が存在することを確認して修正を行ったものである．LLMBar データセットは 2 名のアノテータによる回答評価の一致率が 90% を超えていると報告されており，評価の信頼性が高いことが示されている．さらに，Zeng らが LLMBar との比較を目的として既存のデータセットを加工して作成した Processed (FairEval [9], LLMEval^2 [10], MT-Bench [11]) も使用した．Processed は，LLMBar と公平な比較を行うため，英語以外の質問や回答，または同点 (“TIE”) となった例を全て削除し，必要に応じてタスクの説明を追加するなどの処理を施したものであ

表 1: 生成されたチェックリストの統計情報. GPT-4o と Claude 3.5 Sonnet によって生成された数を合算して表示している.

Stat	Baseline	count_3	count_5	count_10	Metric	detail	detail++
Min	2	0	0	0	0	0	0
Max	19	4	6	20	27	30	31
Avg	6.35	2.95	4.91	9.44	12.01	9.58	10.83
Sum	10843	5,206	8,652	16,628	20,900	17,137	19,104

る．本実験では，これら 8 種類のデータセット，計 885 問を使用した．全てのデータセットには，各質問に対して 2 つの回答が存在し，「1」または「2」のラベルを使って，どちらの回答がより良いかを選択するペアワイズ評価が行われている．

回答の評価には GPT-4o，Qwen2.5-32B-Instruct，Qwen2.5-7B-Instruct の 3 つのモデルを使用した．評価モデルはまず各チェックリストに対して「yes」，「no」，および該当のチェックリストを使用しない場合の「n/a」のいずれかのラベルを出力し，最終的に総合的な評価ラベルとして「1」または「2」のラベルを出力した．しかしチェックリストによる評価結果が同等であった場合，最終評価ラベルとして「1,2」や空白を出力してしまう場合があった．そのため，チェックリストによる評価が同等な場合は，改めて回答全体の内容を精査し，より適切と思われる回答を評価モデル自身に選択させた．また，出力結果が適切なラベルに該当しない場合は最大 3 回まで再評価を行った．

3.2 チェックリスト生成の結果・分析

表 1 に生成されたチェックリストの統計情報を示す．全 885 問に対して，合計 98,470 項目のチェックリストが生成された．項目数に制限を設けた Count では，79%のケースで指示通りに項目数の上限を守ることができていた．一方，項目数制限のない Metric，Detail では，平均 10 項目が出力されていた．また detail と detail++ においては，同じデータセットにも関わらず，項目数がチェックリストを生成させるモデルによって最大 2 倍以上異なるケースが観測された．detail において過剰にチェックリストが生成されているケースを確認したところ，言い回しが異なるだけで趣旨は同じ項目が複数存在していた．こうしたケースについてはチェックリストの整理を行うよう修正を行わせることが有効である可能性がある．それに対し，detail++は予想される回答に対してよりその内容に踏み込んだ項目を生成するよう設計されている．そのため，回答が一意に定まらないオープンエンドの質問では，LLM による質問の解釈の自由度が上がり，その結果として生成された項目数に差が生じる可能性が示唆された．

表 2: 各評価モデルにおいてチェックリストなしの一致率を上回った回数が多かった上位 3 手法．どのモデルでも共通して項目数制限（count_n）と付加情報（detail++）が含まれており，これらの生成手法が有効であることが示唆される．

順位	GPT-4o	Qwen2.5-32B	Qwen2.5-7B
1	count_3	detail	detail++
2	detail++	detail++	count_10
3	count_5	count_3	count_3

次にチェックリストの内容の定性分析を行った．Count では，不必要にも関わらず指定された上限の項目数までチェックリストを生成し，質問に関連性の低い項目が含まれるケースが散見された．Metric においては，数学的な質問に対して共感性を問うなど，指定された 13 の評価指標全てに関連する項目を生成しようとしてしまう傾向が見られた．一方，detail では「回答は正確であるか」といった基本的かつ表面的な内容に焦点が当てられており，detail++では表面的な項目に加え，質問への具体的な答えや深い内容に踏み込む項目が含まれていた．

3.3 評価実験の結果・分析

チェックリスト生成手法 6 種（Baseline を除外した count_3，count_5，count_10，Metric，detail，detail++），チェックリスト生成用モデル 2 種，評価データセット 8 種，評価用モデル 3 種の組み合わせである 288 ケースのうち 65 ケース（22.6%）で，各評価モデルがチェックリストを使用しない場合 (以下 no checklist と呼ぶ) を上回る一致率が得られた．この結果よりチェックリストの有効性は部分的なものに留まることが示唆された．

しかし，表 2 に示すように Count および detail++ の適用が評価モデルを問わず有効であった．またベースラインと比較すると，提案手法によって生成したチェックリストを用いることで，43.8%の割合で一致率が向上した．そして，各評価モデルが no checklist の性能を上回ったケースの割合は，GPT-4o では 36.9%であったのに対し，Claude 3.5 Sonnet では 63.1%に達した．このことより，評価モデルの選定に加えて，チェックリスト生成モデルの選定も評価

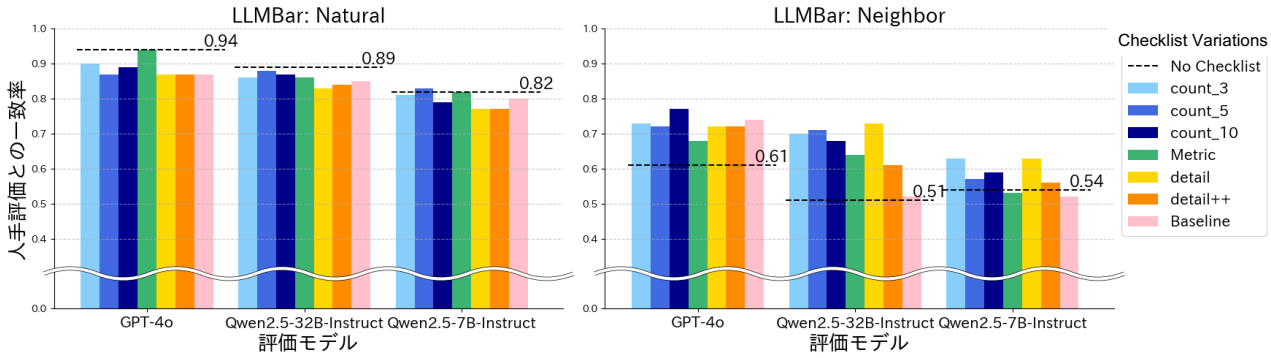

図 2: 各評価モデルと生成方法におけるチェックリスト使用時の一致率の差分を示した結果．データセットは Natural と Neighbor を使用し，チェックリスト生成には Claude 3.5 Sonnet を採用した．

結果に影響を与えることが分かった．

チェックリスト生成手法別の一致率　評価モデルを GPT-4o であるとき，count_3 は no checklist やベースラインを上回る傾向にあった．一方で，Qwen2.5-32B-Instruct では detail が，Qwen2.5-7B-Instruct では detail++ がそれぞれ 20%以上の割合でベースラインを上回った．これらの結果から，高性能な LLM では項目数の制限が有効であり，モデルの規模が小さくなるにつれて，チェックリストの内容の詳細さが効果的であることが示唆される．

評価データセット別の一致率　評価データセットによって，特に Natural と Neighbor において，チェックリストが一致率に与える影響に顕著な差が見られた．図 2 に Natural と Neighbor に対する，各評価モデルとチェックリスト生成手法別の一致率を示す．人間の嗜好に基づく Natural データセットにおいては，no checklist の場合，今回使用した全てのデータセットの中で最も高い一致率であった．もとの一致率が非常に高かったことも影響してか，評価モデルおよび生成手法に関わらず，ほぼ全ての場合で no checklist と比較して一致率が低下した．一方，LLM を誤解させるよう意図的に設計された Neighbor では，no checklist で 3 モデル全てで今回使用したデータセットの中で一致率が最も低い一致率であった．しかし，チェックリストを使用することでほぼ全ての場合で no checklist と比較して一致率が向上し，最も大きな向上幅は 22%（51% →73%）の向上を示した．この結果より，チェックリストは誤解を招く可能性のあるデータセットにおいて特に有効であることが明らかとなった．

評価モデル別の一致率　同一データセット内で比較すると，評価モデルの性能の良さに応じて一致率が向上する傾向が見られた．また，Neighbor に対してチェックリストを用いて Qwen2.5-7B-Instruct で評価した場合，9%のケースで GPT-4o の no checklist の結果と同等以上の性能を達成した．この結果から，LLM を誤解させるよう意図的に設計されたデータセットであっても，小規模な評価モデルに対してチェックリストを使用することで評価基準を明確化でき，性能の高い LLM と同等以上の評価結果を達成できる可能性を示している．

4　おわりに

本研究では，LLM を用いた自動評価における評価基準の曖昧さを解消するため，6 種類のチェックリスト生成手法を比較検討し，回答の評価を行った．実験の結果，チェックリスト未使用の一致率と比較して，一致率が向上したケースは 22.6%であり，チェックリストの有効性は部分的なものに留まることが明らかになった．一方で，項目数制限やチェックリストの内容の詳細性に着目した生成方法は 3 つの評価モデル全てで有効である傾向が確認された．また，評価モデルが小規模であっても，適切に生成されたチェックリストを使用することで，性能の高いモデルと同等以上の評価結果を得られる可能性があることが示唆された．今後はチェックリスト生成モデルや評価モデル，評価データセットのさらなる拡充を図るとともに，提案したチェックリスト生成手法を相互に関連づけた生成手法の開発およびその評価を進める予定である．

謝辞

本研究は JST 創発的研究支援事業 JPMJFR232R の支援を受けたものです．また，本研究成果は，データ活用社会創成プラットフォーム mdx を利用して得られたものです．

参考文献

[1] Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, and Pilsung Kang. Checkeval: Robust evaluation framework using large language model via checklist, 2024.

[2] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating Instruction Following Ability in Large Language Models. 2024.

[3] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. **arXiv e-prints**, pp. arXiv–2406, 2024.

[4] OpenAI. GPT-4 Technical Report, 2024.

[5] AI Anthropic. Claude 3.5 sonnet model card addendum. **Claude-3.5 Model Card**, Vol. 3, , 2024.

[6] Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, and Shanghaoran Quan. Qwen2.5 Technical Report. 2024.

[7] Jonathan Cook, Tim Rocktäschel, Jakob N. Foerster, Dennis Aumiller, and Alex Wang. TICKing All the Boxes: Generated Checklists Improve LLM Evaluation and Generation. **ArXiv**, 2024.

[8] Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In **International Conference on Learning Representations (ICLR)**, 2024.

[9] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 9440–9450, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[10] Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. Wider and Deeper LLM Networks are Fairer LLM Evaluators, 2023.

[11] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, 2023.

A　生成されたチェックリスト

表 3: MT-Bench データセットにおける各生成方法でのチェックリスト．生成モデルは GPT-4o を使用した．Metric と detail は紙面の都合上，一部省略をした．

Input	$x + y = 4z$, $x \cdot y = 4z^2$, express $x - y$ in z
Baseline	Does the response correctly identify and use the given equations $x + y = 4z$ and $x \cdot y = 4z^2$? Does the response provide a valid algebraic manipulation to express $x - y$ in terms of z? Is the expression for $x - y$ presented clearly and concisely? Are all assumptions or steps clearly stated or logically inferred from the equations provided? Does the response refrain from introducing unnecessary information not relevant to expressing $x - y$ in terms of z?
count_3	Is the generated text relevant to the input prompt? Does the generated text maintain coherence and consistency? Is the length of the generated text appropriate for the task?
count_5	Does the generated output maintain coherence and logical consistency throughout the text? Is the generated content relevant to the provided prompt or input context? Does the generated output adhere to grammatical and syntactical standards? Is the output free from any biased or inappropriate language? Is the response creatively original and not directly copied from the input or any known sources?
count_10	Does the generated output address all key components of the input prompt? Is the generated content relevant to the domain or topic specified in the input? Does the output maintain consistency in style, tone, and structure with the input context? Is the information in the generated output factually correct based on known data? Does the output avoid repetition of ideas or phrases? Is the language used in the output coherent and contextually appropriate? Are there any logical inconsistencies or contradictions within the output? Does the output follow any specified constraints or guidelines provided in the input (such as word count, format, etc.)? Is the output free from grammar and spelling errors? Does the generated content maintain focus on the specified task without deviating unnecessarily?
Metric	Is the generated text grammatically correct? Does the text flow naturally without awkward pauses or constructions? Are the details presented in the generated text consistent throughout? Is the tone of the text suitable for the intended audience? Does the text cover all necessary aspects of the topic? Does the text accurately represent factual information? Is the generated content useful for the intended purpose or task? Is the generated text free from bias or prejudice?
detail	Is the response directly related to the input prompt? Does the response address the specific requirements or questions posed by the input? Is the information provided in the response factually correct? Is the response logically coherent and consistent within itself? Does the response fully address all aspects of the input prompt? Is the input type (e.g., mathematical, narrative, informational) clearly identified?
detail++	Does the answer use the given equations $x + y = 4z$ and $x \cdot y = 4z^2$? Is the answer expressed in terms of z? Does the solution involve solving for x and y separately? Is algebraic manipulation used to combine the equations? Does the answer include steps showing how $x - y$ is derived? Is the final expression for $x - y$ simplified? Are any assumptions made about z being non-zero? Is the solution mathematically consistent throughout? Does the answer consider both positive and negative roots if applicable? Is there a verification step to check if the derived expression satisfies the original equations?