真面目 LLM と不真面目 LLM で推論能力は変わるか？

堀尾海斗　河原大輔
早稲田大学理工学術院
{kakakakakakaito@akane.,dkw@}waseda.jp

概要

本研究では、真面目さという性格を LLM に付与した場合のタスク正解率への影響を検証する。手法としては、真面目さ、不真面目さを付与する複数種類のプロンプトを用いる。実験では、日本語の 3 種類のタスクを使用して各プロンプトを検証する。実験の結果、真面目さを付与した場合はプロンプトによって正解率が向上することもあれば低下することもあること、不真面目さを付与した場合は基本的に正解率が低下することがわかった。また、正解率に大きな変化を与える言語表現が存在することも判明した。

1　はじめに

大規模言語モデル (LLM) は日々、様々な用途で使用されており、生成精度のさらなる向上が求められている。LLM の生成精度の向上手法は、ファインチューニングによるパラメータの最適化やプロンプト手法の改善など多様に存在する。

プロンプト手法の改善の代表例には、考えの過程を明記させる Chain-of-Thought Prompt [1] や役割を与える Role-Play Prompt [2]、LLM の感情に訴えかける EmotionPrompt [3] がある。これらの研究で扱われている役割と感情は元来、人間の発話を変化させる要素であるが、これら以外に主要な要素として性格がある。本研究では、性格の一つである真面目さ、不真面目さを LLM が理解して生成を変化させることができるかを、Role-Play Prompt の手法を参考に日本語で検証する。真面目さ、不真面目さを付与するプロンプトを、レタータスク、 コインタスク、オブジェクトタスクの 3 つで実験し、タスク正解率を比較する。レタータスクは既存のデータセットを使用し、コインタスクとオブジェクトタスクは英語のデータセットを参考に、新たに日本語で作成する。

実験の結果、真面目さを付与するプロンプトは、その内容によっては正解率が向上するものも有れば、低下するものも有り、その程度もタスクで異なることがわかった。不真面目さを付与するプロンプトは基本的に正解率を落とすが、正解率の下がり幅もタスクやプロンプトの内容で異なることがわかった。また、モデルの生成を大きく変化させる言語表現が存在することも判明した。これらの結果から、モデルは常にプロンプトに従うものではないことが判明し、適切なプロンプト設計が必要になることがわかった。

[2]

2　関連研究

2.1　様々なプロンプト手法

モデルの生成精度を向上させるプロンプト設計の手法には、Role-Play Prompt [2] や、EmotionPrompt [3] などがある。Role-Play Prompt とは、特定の役割になりきる旨の指示をタスクの前に提示するプロンプトで、EmotionPrompt とは精度を上げることを感情的に促すプロンプトである。Role-Play Prompt の手法は本研究で用いる為、2.2 節で詳細に述べる。

LLM に対して性格を付与させるようなプロンプト設計を含む先行研究 [4, 5, 6] が存在する。これらは LLM に性格が付与されるかどうかを評価しているが、タスクに対する精度の変化は評価していない。

プロンプトの構造を大きく変化させずに、プロンプトの始めにスペースや、挨拶を入れるだけなどの小さな変化でもモデルの生成に変化が生じることや、プロンプトの入力長やタスクの選択肢の順序によって生成精度が変化することもわかっている [7, 8, 9]。

2.2　Role-Play Prompt

Role-Play Prompt とは、モデルに与えるプロンプト設計の手法の 1 つである。Kong ら [2] は、Role-Play Prompt でモデルにタスクを回答させるまでの流れと

表 1　Role-Play Prompt の例

通常プロンプト	ユーザ：問題に回答してください。 モデル：はい。問題に回答します。 ユーザ：「」で答えてください。＋ [改行] ＋ { 問題 } ＋ [改行] ＋ 解答：
Role-Play Prompt	ユーザ：これからあなたは真面目な学生になります。 あなたは真面目なので問題には正解しますし、しっかり答えます。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。 ユーザ：「」で答えてください。＋ [改行] ＋ { 問題 } ＋ [改行] ＋ 解答：

表 2　タスク例

	問題	正解
レター	「池田 博人」の姓、名のそれぞれの最後の一文字を姓、名の順で繋げてください。	田人
コイン	コインは現在、表です。濱田さんがコインを裏返します。中田さんはコインを裏返しません。中間さんはコインを裏返しません。桐山さんがコインを裏返します。コインはまだ、表ですか？ここで裏返すとは表と裏の反転を意味します。	はい
オブジェクト	田中さん、佐藤さん、山下さん、山田さんはゲームをしています。ゲーム開始時に、田中さんはピンク色のボールを、佐藤さんは赤色のボールを、山下さんは白色のボールを、山田さんは黄色のボールを持っています。ゲームが進むにつれてプレーヤーは持っているボールを交換します。まず田中さんと佐藤さんがボールを交換します。その後、山下さんと山田さんがボールを交換します。最後に田中さんと山下さんがボールを交換します。ゲームの終わりに田中さんは何色のボールを持っていますか？	黄色のボール

して 3 段階で実験を行っている。

まず 1 段階目として、ユーザ側からモデルに対して特定のロールに従う旨のプロンプトを与える。2 段階目としてモデルが、与えたプロンプトに対して、ロールを理解し、特定のロールを全うする内容の返事を行う (この部分も我々が定める)。3 段階目として、ユーザ側がモデルにタスクを出題する。Role-Play Prompt の例を表 1 に示す。先行研究では、様々なロールで実験されており、不注意な学生よりも数学の教師のロールを与えた時の方がより良い精度を出している。

3　タスクとプロンプトの設計

3.1　タスクの設計

本研究では、Role-Play Prompt 論文 [2] と同様に、レータスク、 コインタスク、オブジェクトタスクの 3 つのタスクを用いて日本語で実験を行う。これら 3 つのタスクの例を表 2 に示す。

3.1.1　レータスク

レータスクは、人名の姓と名の最後の文字を抜き取って、2 文字に繋げたものを答えるタスクである。レータスクのデータセットは、既存のもの [10] を使用する。

3.1.2　コインタスク

コインタスクは、始めは表の状態のコインを、4 人の人物が動かすことによって最終的にコインが表を向いているか裏を向いているか答えるタスクである。コインタスクのデータセットは、英語のもの [1, 11] を参考に、日本語で作成する。

3.1.3　オブジェクトタスク

オブジェクトタスクは、登場人物が始めから物を持っており、登場人物同士で持っている物を交換した結果、最終的に特定の人物が何を持っているかを答えるタスクである。オブジェクトタスクのデータセットは、英語のもの [12] を参考に、日本語で作成する。

3.2　プロンプトの設計

本研究の実験で使用するプロンプトを作成する。性格を付与せず、問題の回答を促すだけのプロンプトを 1 種類、真面目さを付与するプロンプトを 6 種類、不真面目さを付与するプロンプトを 9 種類、人手で作成する。以降、真面目さを付与するプロンプトを真面目プロンプト、不真面目さを付与するプロンプトを不真面目プロンプトと表す。性格が同じプロンプトでも複数種類作成しているのは、同じような言い回しでも文の順番や同義の単語の置き換え、タスクの説明の有無によって生成に変化が起きるかを検証する為である。問題の回答を促すだけのプロ

表 3　真面目プロンプト (レター)

1	ユーザ：これからあなたは真面目な学生になります。あなたは真面目なので問題には正解しますし、しっかり答えます。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。
2	ユーザ：これからあなたは真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは真面目なので問題には正解しますし、しっかり答えます。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。
3	ユーザ：これからあなたは真面目な学生になります。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。真面目な学生ということなので問題には正解をします。
4	ユーザ：これからあなたは真面目な学生になります。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。真面目な学生ということなので問題の正答率を意図的に上げます。
5	ユーザ：これからあなたは真面目な生徒になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは真面目なので問題には正解しますし、しっかり答えます。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。
6	ユーザ：これからあなたは真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは適切に答えます。 モデル：もちろん！真面目な学生として問題に回答します。どうぞご質問ください。

ンプトは表 1 の通常プロンプトである。レタータスクで用いる真面目・不真面目プロンプトを、表 3、表 4 に示す。

4　実験

4.1　実験設定

実験に使用する LLM は GPT-4 とし、ハイパーパラメータは、max-tokens を 512、temperature を 0 とし、生成回数は 1 回とする。

通常プロンプト、6 種類の真面目プロンプト、9 種類の不真面目プロンプトを用いて、レタータスク、コインタスク、オブジェクトタスクをモデルに解かせる。3 つのタスクをそれぞれ 500 問解かせてその正答率を測る。プロンプトをモデルに与える際の形式は表 1 の形式である。

4.2　評価方法

通常、真面目、不真面目の 3 種類のプロンプトで得た生成を評価する。

レタータスクの生成は、生成に含まれる最後の鉤括弧 (「」) に含まれる表現が、正解と完全一致しているかで自動評価する。

コインタスクは、「はい」か「いいえ」で答えられるタスクであるが、モデルの生成が「裏です」や「表でも裏でもない」といった「はい」、「いいえ」以外で回答する現象が見られた。その為、コインタスクの生成の評価は、生成の最後の鉤括弧に含まれる表現の内、「表」と「裏」の出現回数が多い方をモデルの回答とし、表なら「はい」、裏なら「いいえ」に変換して、正解と完全一致しているかどうかで評価する。なお、表と裏の出現回数が同じの場合は不正解とする。

オブジェクトタスクは、生成に含まれる表現の中に、正解の色が含まれているかどうかで評価する。

4.3　実験結果・考察

真面目・不真面目プロンプトを用いた 3 つのタスクの正解率を表 5、表 6 に示す。コインタスクにおいて通常、不真面目プロンプトで不正解となる問題が真面目プロンプトで正解になった例を付録の表 7 に示す。

真面目プロンプトを用いた結果、レター, コイン, オブジェクトタスクの全てのタスクで、プロンプトの違いによっては正解率が向上するものと低下するものがあった。不真面目プロンプトを用いた結果、レター, コイン, オブジェクトタスクの全てのタスクで正解率が低下した。プロンプトの違いによって、正解率の低下度合いには差があり、通常プロンプトの生成との差が 2 %以下のものもあれば、通常プロンプトに比べて、50%以上正解率が低下しているものもある。

また、不真面目プロンプトにおいて一際正解率が低下しているものが確認された。正解率が低いもののプロンプトに注目すると、「真面目に答えません。」という言語表現がプロンプト中の 2 文目以降

表 4 不真面目プロンプト (レター)

1	ユーザ：これからあなたは不真面目な学生になります。あなたは不真面目なので問題にはミスをしますし、真面目に答えません。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
2	ユーザ：これからあなたは不真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは不真面目なので問題にはミスをしますし、真面目に答えません。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
3	ユーザ：これからあなたは不真面目な学生になります。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。不真面目な学生ということなので問題にはミスをします。
4	ユーザ：これからあなたは不真面目な学生になります。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。不真面目な学生ということなので問題の正答率を意図的に下げます。
5	ユーザ：これからあなたは不真面目な生徒になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは不真面目なので問題にはミスをしますし、真面目に答えません。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
6	ユーザ：これからあなたは不真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたはテキトーに答えます。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
7	ユーザ：これからあなたは不真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。あなたは不真面目なので問題にはミスをしますし、不真面目に答えます。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
8	ユーザ：これからあなたは不真面目な学生になります。あなたは不真面目なので問題にはミスをしますし、真面目に答えません。単語の最後の文字を連結して新しい単語を取得する問題を解いています。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。
9	ユーザ：あなたは不真面目なので問題にはミスをしますし、真面目に答えません。これからあなたは不真面目な学生になります。単語の最後の文字を連結して新しい単語を取得する問題を解いています。 モデル：もちろん！不真面目な学生として問題に回答します。どうぞご質問ください。

表 5 真面目プロンプトのタスク正解率

	レター	コイン	オブジェクト
通常	0.940	0.588	0.454
1	0.934	0.608	0.424
2	0.848	0.566	0.446
3	0.944	0.620	0.418
4	0.944	0.612	0.442
5	0.866	0.572	0.432
6	0.860	0.566	0.500

表 6 不真面目プロンプトのタスク正解率

	レター	コイン	オブジェクト
通常	0.940	0.588	0.454
1	0.860	0.036	0.000
2	0.828	0.124	0.002
3	0.898	0.576	0.314
4	0.924	0.574	0.328
5	0.840	0.036	0.004
6	0.810	0.548	0.048
7	0.820	0.336	0.008
8	0.750	0.010	0.010
9	0.756	0.560	0.436

に含まれているという共通点があった。同じ不真面目プロンプトでも、ただ誤りを生成しているものもあれば、明らかに真面目に回答する気が無いような生成をさせるプロンプトがあった。コインタスクのプロンプトの違いによる生成の違いを表 8 に示す。

実験結果から、性格を付与するプロンプトの有効性は確認され、プロンプトのモデルへの影響は大きいことがわかった。また、性格を付与するプロンプトにはモデルへ大きな影響を与える言語表現が存在することがわかった。

5 おわりに

本研究では、Role-Play Prompt の手法を用いて LLM への真面目、不真面目の性格を付与し、タスク精度の検証を行った。性格を付与するプロンプトを用いる場合は、特定の言語表現がモデルの性格理解を向上させることも判明した。

展望として、本研究以外のタスクでの実験や、プロンプトにてモデルの性格理解を向上させる言語表現の探索が挙げられる。

謝辞

本研究は JSPS 科研費 JP24H00727 の助成を受けて実施した。

参考文献

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 24824–24837. Curran Associates, Inc., 2022.

[2] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. Better zero-shot reasoning with role-play prompting. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 4099–4113, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[3] Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli. arXiv, 2023. abs/2307.11760.

[4] Jen tse Huang, Wenxuan Wang, Man Ho Lam, Eric John Li, Wenxiang Jiao, and Michael R. Lyu. Revisiting the reliability of psychological scales on large language models. arXiv, 2023. abs/2305.19926.

[5] Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. Evaluating and inducing personality in pre-trained language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 10622–10643. Curran Associates, Inc., 2023.

[6] Greg Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. Personality traits in large language models. arXiv, 2023. abs/2307.00184.

[7] Abel Salinas and Fred Morstatter. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 4629–4651, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[8] Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. 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. 15339–15353, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Findings of the Association for Computational Linguistics: NAACL 2024**, pp. 2006–2017, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[10] 堀尾海斗, 村田栄樹, 王昊, 井手竜也, 河原大輔, 山崎天, 新里顕大, 中町礼文, 李聖哲, 佐藤敏紀. 日本語における Chain-of-Thought プロンプトの検証. 人工知能学会全国大会論文集, 2023.

[11] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 22199–22213. Curran Associates, Inc., 2022.

[12] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, and Adrià Garriga-Alonso et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv, 2023. abs/2206.04615.

A　プロンプトによる生成結果の変化の例

表 7 には 3 種類のプロンプトの変化による生成の変化の例を示す。表 8 にはコインタスクにおける不真面目プロンプトの言語表現の違いによる生成の変化例を示す。

表 7　プロンプト種類の変化による生成例

	問題：コインは現在、表です。城島さんがコインを裏返します。坂本さんがコインを裏返します。 長野さんはコインを裏返しません。木村さんはコインを裏返しません。コインはまだ、表ですか？ ここで裏返すとは表と裏の反転を意味します。
通常	「裏です」
真面目	「はい」
不真面目	「宇宙人が持っていきました」

表 8　不真面目プロンプト内の言語表現の違いによる生成の変化例

プロンプト 1	「宇宙にはコインなんて存在しないんだよ」 「もちろん、コインは宇宙に行ってしまいました。」
プロンプト 4	「いいえ」 「はい」