ヒューリスティックと遺伝的アルゴリズムを用いた自動プロンプトチューニング手法

進藤稜真　ジェプカ・ラファウ　竹下昌志　荒木健治
北海道大学
{shinto.ryoma, takeshita.masashi.68}@gmail.com
{rzepka,araki}@ist.hokudai.ac.jp

概要

良質なプロンプトを作成することは大規模言語モデルの性能を引き出すために重要であるが，プロンプト作成には人間の手間と時間的コストを必要とする．そこで本研究では，一般に広く用いられているプロンプトに関するヒューリスティックと遺伝的アルゴリズムを用いて，タスクに特化したプロンプトを自動でチューニングする手法を提案する．評価実験により，提案手法でチューニングしたプロンプトは人手で作成したプロンプトと同水準かそれ以上の性能を持つことを確認した．

1　はじめに

高度な推論能力を持つ大規模言語モデルの台頭とともに，その性能を引き出すためのプロンプトに関する研究も増加している．モデル自身に思考過程を出力させる Chain-of-Thought(CoT)[1] や，Zero-shot で CoT を行う Zero-shot CoT[2] がその代表例である．また，“Take pride in your work and give it your best.” などの感情刺激を引き起こすプロンプト [3] や，“あなたはプロのクイズプレイヤーです．” と役割を明示するプロンプト [4] など，モデルの性能を引き出すためのヒューリスティックも多数存在する．

このように，適切なプロンプトの設計により LLM のタスクに対する性能を向上させることができるが，作成する際の問題点として人間の手間と時間的コストがかかることが挙げられる．また，前述したヒューリスティック [3, 4] 以外にも，細かなニュアンスの違いやプロンプトの入力順によってモデルの出力は変わるため [5]，良質なプロンプト作成のために考慮すべき構成要素の組み合わせ数は多い．

以上の問題点を解決する手法として，本稿ではヒューリスティックと遺伝的アルゴリズム [6] を用いた自動プロンプトチューニング手法を提案する．提案手法では，モデルへの入力プロンプトを 1 個体，意味単位で分割した各モジュールを遺伝子とみなし遺伝的アルゴリズムを適用する．これにより，ヒューリスティックに加えて細かなニュアンスやモジュールの順番も探索しつつ，タスクに特化したプロンプトの自動チューニングが可能となる．

評価実験では，自動チューニングしたプロンプトを JGLUE[7, 8] の JCommonsenseQA と JNLI を用いて評価し，自然言語処理を専攻する学生の作成したプロンプトと同等かそれ以上の正答率を実現した．

2　関連研究

プロンプトの自動チューニング手法には，大きく分けて 2 通りのアプローチが存在する．

1 つ目が，Soft-Prompting[9, 10] や，LLM の出力を解答ラベルとして Fine-Tuning を行う手法 [11, 12] など，勾配やパラメータの更新が必要な手法である．しかし，モデルが大きくなるにつれ勾配の計算コストは高くなり，また，モデルへのアクセスが API に限定される場合もあるため，実用性に乏しい．

対して，2 つ目は直接プロンプトを操作して最適化を目指すパラメータの変更が必要ない手法 [13] である．LLM 自身が出力にフィードバックを行いプロンプトを改善する手法 [14, 15] などがある．提案手法もこのアプローチをとっており，モデルを問わずチューニングできる実用的なメリットがある．

提案手法で用いた遺伝的アルゴリズム [6] は，生物の進化のメカニズムを参考に考案された，近似解を求めるアルゴリズムの一つである．解の初期集団を生成し，適応度に基づいて交叉，突然変異のプロセスを繰り返すことで，より良い解の探索を行う．

この遺伝的アルゴリズムを用いてプロンプトを自動チューニングする手法には，GPS[16] と

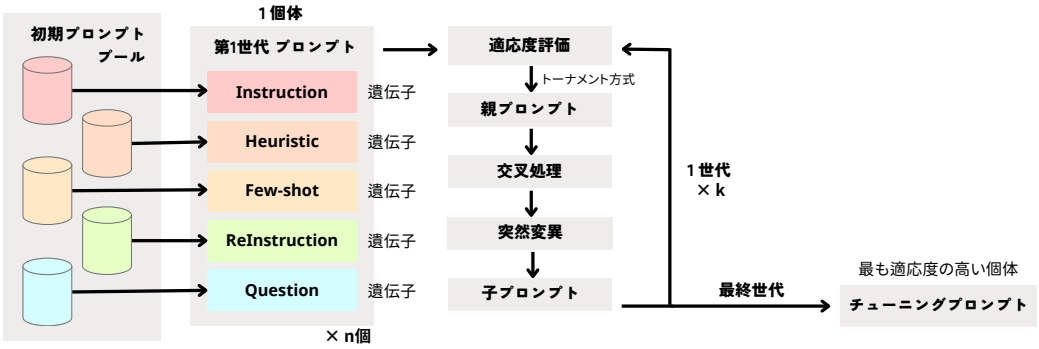

図 1　提案手法の概略図．初期プロンプトプールから生成した n 個の第 1 世代プロンプトに対し，適応度評価・交叉・突然変異のプロセスを k 世代繰り返し，最終世代において最も適応度の高いプロンプトをチューニングプロンプトとする．

PromptBreeder[17] がある．提案手法も上記 2 つと同様に遺伝的アルゴリズムを応用した手法であるが，ヒューリスティックを初期値として明示的に取り入れることで，それらを基にした多様なプロンプトの探索を目指す．また，新たなヒューリスティックが報告された場合には初期プロンプトとして追加するだけでよく，柔軟な拡張性を持つ．

3　提案手法

図 1 に提案手法の概略図を示す．提案手法では，入力プロンプトを 5 つのモジュール（Instruction, Heuristic, Few-shot, ReInstruction, Question）に分割しチューニングを行う（3.1 節）．

まず，人手で作成した初期プロンプトを各モジュールごとにランダムに選出し，それらを結合してモデルに入力する 1 つのプロンプトを生成する．この操作によって n 個のプロンプトを作成した後，それぞれの適応度を評価し（3.2 節），トーナメント方式を用いて次世代の親プロンプトを n 個用意する．これらに交叉処理（3.3 節）と突然変異（3.4 節）を行うことで，子プロンプトを生成する．

以上の手順を 1 世代として合計 k 世代繰り返し，最終世代の n 個の中で最も適応度が高いプロンプトを最終的なチューニングプロンプトとする．

提案手法と遺伝的アルゴリズムの対応関係は，1 つの入力プロンプトが 1 個体，各モジュールのプロンプトが遺伝子に対応する．適応度・交叉処理・突然変異については以下に詳細を示す．

3.1　Prompt モジュール

モデルに入力されるプロンプトは以下の 5 つのモジュールからなる [1)]．

Instruction　モデルが取り組むタスクの指示を行う．初期プロンプトには，Zero-shot CoT を促すプロンプトも一部含まれている．

Heuristic　モデルの性能を引き出すヒューリスティックプロンプトを与える役割を担う．主に [3] や [4] で用いられた英語のプロンプトを日本語訳し，初期プロンプトとして用意した．

Few-shot　モデルに与えるタスクと同形式で例題を提示する．Few-shot 数はベースラインに合わせて 3 に設定した．初期プロンプトは，Train データセットからランダムにサンプリングを行う．

ReInstruction　モデルに対し再度タスクの指示を明記する役割を担う．ReInstruction は省略することが可能である．

Question　モデルに与えるタスクを保持するモジュールである．適応度評価や評価実験の際に，Question モジュールの内容を 1 問ずつ更新してモデルに入力する．

3.2　適応度評価

適応度を「個体（入力プロンプト）のタスク正解率」と定義し，各個体がどれだけタスクに対して有効であるかを定量的に評価する．そこで，train データセットからランダムに 50 例サンプリングし，その正解率を適応度とする．

また，この適応度に基づき親プロンプトを生成する．親プロンプトの決定方法は，各世代でランダムに 2 個体を選び，適応度の高い個体で低い個体を上書きする「トーナメント方式」を採用した．

3.3　交叉処理

親プロンプトをランダムに 2 個体選出し，それぞれのモジュールごとに 50 %の確率で互いの遺伝子

1)　初期プロンプトの具体例は，A.1 を参照されたい．

（各モジュールのプロンプト）を入れ替える「一様交叉」を行う．これにより，適応度の高い個体の性質を備えた子プロンプトが生成される．Few-shot モジュールでは，各例が個別で交叉の対象となる．

3.4 突然変異

突然変異には「再選択」,「言い換え」,「順番変更」の 3 つの方法がある．

再選択 モジュールごとに現在の遺伝子を新たに初期プロンプトから選択した遺伝子で上書きする．これにより，最初の世代で個体の遺伝子として選ばれなかった初期プロンプトの探索が可能になる．

言い換え Few-shot と Question を除いたモジュールにおいて，GPT-4[18] を用いてプロンプトを言い換える（A.3 参照）．「言い換え」は，意味を変えずに異なる言語表現を得られるので，ニュアンスの違いによるモデルの性能の変化を探索するために行う．

順番変更 各モジュールの順番をランダムに組み替えることで，タスクに対し効果的なモジュールの順番を探索する．初期プロンプトは，モジュールの順番によらず意味が通るように設計されている．

4 実験

4.1 提案手法の有効性の評価実験

提案手法によって自動チューニングしたプロンプト（A.2 参照）の性能を評価するため，自然言語処理を専攻する大学生・大学院生の作成したプロンプトとの比較を行った．学生が 3 名ずつ各タスクに対してプロンプトを作成し，3 回解いた結果から平均正答率を求めた（詳細 A.4）．提案手法も同様に，チューニングしたプロンプトで 3 回タスクを解き，その平均正答率を求めた．ベースラインにはモデル開発元の公開スコア[2)]を設定した．チューニングは個体数 n を 10，世代数 k を 30 と設定して行った．

使用したモデルは，同等サイズの日本語 LLM で最高水準の性能を持つ Japanese Stable LM-3B-4E1T Base [3)], Japanese Stable LM Base Gamma 7B [4)]である．また，評価タスクには日本語理解ベンチマーク JGLUE[7, 8] から，JGLUE の他タスクと比較してベースラインスコアの低い JCommonsenseQA(JCSQA) と JNLI を選択した．これは，提案手法のチューニングによる改善の余地があると判断したためである．

表 1 モデル・タスク別の平均正答率．Baseline : モデル開発元の公開スコア．Human : 自然言語処理を専攻する学生の作成したプロンプト．Ours : 提案手法．

Model	Task	Baseline	Human	Ours
3B	JCSQA	**36.01**	27.99	26.15
3B	JNLI	**51.27**	37.23	45.75
7B	JCSQA	60.59	82.91	**85.02**
7B	JNLI	20.58	37.15	**47.99**

4.2 突然変異率の妥当性の検証実験

本研究では，突然変異が多様なプロンプトの探索に重要な要素であると考える．そこで，ハイパーパラメータとして設定した突然変異率 0.1 の妥当性を議論するため，突然変異率を 0.0 から 0.3 まで変更し適応度の推移を観察した．タスクにはベースラインの高い JCSQA を選択し，モデルには Stability AI Stable LM Base Gamma 7B[4)] を用いた．

5 結果

5.1 提案手法の有効性の評価実験結果

提案手法と人手で作成したプロンプトによる各タスクの平均正答率，ベースラインを表 1 に示す．表 1 から，提案手法は人手で作成したプロンプトと同等かそれ以上の正答率を達成していることが分かる．さらに，7B モデルでは各タスクともにベースラインを約 25 ポイント更新した．

また，図 2，3 には各世代の平均適応度を 30 世代までグラフにプロットし，各タスクにおけるモデルごとのチューニング過程を示す．図 2，3 からは，7B モデルの JCSQA において第 1 世代の平均適応度から約 50 ポイント改善される様子が確認できる．3B モデルでの JCSQA や，両モデルでの JNLI でも，30 世代を経て約 10 ポイントのスコア向上が見られた．

5.2 突然変異率の妥当性の検証実験結果

図 4 に突然変異率を 0.0 から 0.3 まで変更した場合の各世代の平均適応度の推移を示す．図 4 から，突然変異率 0.1 の場合は安定して適応度が上昇する様子が見られる．対して，突然変異率が 0.0 の場合，世代数を重ねても適応度は上昇しない．また，突然変異率を 0.2，0.3 と大きくした場合は適応度の推移が不安定になることが分かる．

2) https://ja.stability.ai/blog/japanese-stable-lm-3b-4e1tjapanese-stable-lm-gamma-7b
3) https://huggingface.co/stabilityai/japanese-stablelm-3b-4e1t-base
4) https://huggingface.co/stabilityai/japanese-stablelm-base-gamma-7b

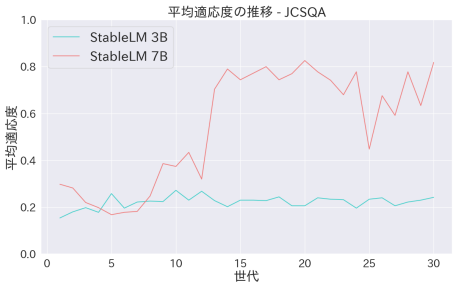

図 2　各世代の平均適応度の推移（JCSQA）

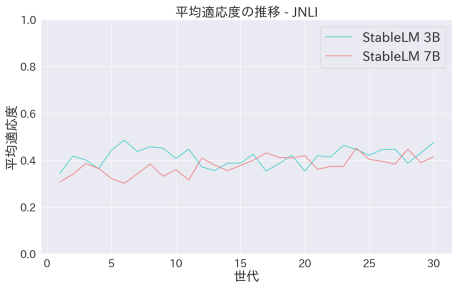

図 3　各世代の平均適応度の推移（JNLI）

6　考察

6.1　提案手法の有効性について

提案手法が人手で作成したプロンプトと同等以上の正答率を達成し（表 1），本研究で用いたモデルにおいて，人間と同レベル以上のプロンプトを自動チューニング可能であることが示唆された．

また，図 2 では 7B モデルで JCSQA を解いた場合に適応度が大きく上昇する様子が見られた．これは，適応度をサンプリングした 50 例による正答率と設定したためだと推測される．Human，Baseline のスコアがともに高く（表 1），7B モデルは JCSQA を解く能力が高いと言える．タスク性能が高いほど，良質な個体と同世代の他個体との適応度の差が明確になり，後の世代に継承されやすくなる．

一方で，3B モデルで JCSQA を解いた場合や両モデルで JNLI を解いた場合，平均適応度の改善は 10 ポイント程度に留まった（図 2，3）．JCSQA は 5 択，JNLI は 3 択タスクであることを考慮しつつ，これらの Baseline や Human スコアを見ると（表 1），ランダムな回答（JCSQA:20%, JNLI:33%）との差が，7B モデルで JCSQA を解いた場合に比べ小さい．よって良質な個体とその他の個体の差が不明確になり，後の世代に継承されにくくなると考えられる．

以上から，提案手法は性能の高い LLM においてより効果的なチューニングが期待でき，GPT-4[18] などの高性能 LLM でも提案手法は有効であると推測される．また，タスク性能が低いモデルにおいても提案手法の有効性を高めるには，適応度評価時のサンプリング数を増やし，確率的な揺れを低減させるなど，モデルに対して効果的なプロンプトが明確になるような改善が必要になると考えられる．

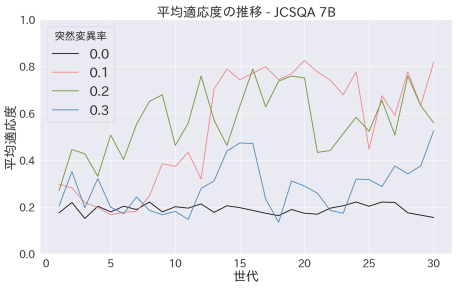

図 4　突然変異率別の平均適応度の推移（JCSQA,7B）

6.2　突然変異率の妥当性について

図 4 より，突然変異率が 0.0 の場合には平均適応度は全く上昇しないことが分かる．これは遺伝的アルゴリズムの性質上，突然変異が起こらなければ第 1 世代の中で最も適応度の高い個体に収束し，その個体の適応度が限界になるためである．この結果から，突然変異が多様なプロンプトの探索において重要な役割を担っていると示唆される．

また，突然変異率を 0.2，0.3 と大きくするにつれて平均適応度は不安定になる．突然変異率を大きくすると，適応度の高い個体も突然変異によって失われる可能性が大きくなり，良質な個体を次世代に継承するのが難しくなることが要因だと推測される．

以上から，ハイパーパラメータの突然変異率 0.1 は探索と継承のバランスを保っており，今回の実験設定においては有効であると考えられる．

7　おわりに

本研究では，プロンプト作成の手間と時間的コストを削減するために，遺伝的アルゴリズムとヒューリスティックを用いたプロンプト自動チューニング手法を提案した．提案手法は，評価実験にて人間の作成するプロンプトと同水準かそれ以上の結果を残し，本研究で用いたモデルにおいて人間と同レベル以上のプロンプトを自動チューニング可能であることを示唆した．今後の課題として，他の様々なモデルやタスクを用いて提案手法の有効性のさらなる検証を行う．また，視覚言語モデルなど画像を含めたマルチモーダルへの拡張も視野に入れたい．

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] 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.

[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 preprint arXiv:2307.11760**, 2023.

[4] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4. **arXiv preprint arXiv:2312.16171**, 2023.

[5] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[6] Tom M Mitchell. The need for biases in learning generalizations. Technical Report CBM-TR-117, Department of Computer Science, Laboratory for Computer Science Research, Rutgers University, 1980.

[7] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[8] 栗原健太郎, 河原大輔, 柴田知秀. Jglue: 日本語言語理解ベンチマーク. 言語処理学会第 28 回年次大会, 2022.

[9] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. **AI Open**, 2023.

[10] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[11] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 1051–1068, Singapore, December 2023. Association for Computational Linguistics.

[12] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 15476–15488. Curran Associates, Inc., 2022.

[13] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In **The Eleventh International Conference on Learning Representations**, 2023.

[14] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In **The Eleventh International Conference on Learning Representations**, 2023.

[15] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[16] Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. GPS: Genetic prompt search for efficient few-shot learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 8162–8171, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[17] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. **arXiv preprint arXiv:2309.16797**, 2023.

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

A 付録

A.1 初期プロンプトの例

Instruction（計 24 文）

”###指示 \n[問題] に対する回答を，[選択肢] の中から選び数字で答えてください。”
”###指示 \n[問題] に対する回答を，[選択肢] の中から選び数字で答えてください。答えの根拠も回答してください。”
”[問題] に対する回答を，[選択肢] の中から選び数字で答えてください。その際，答えの推論過程を記述してください。”
”[問題] に対する回答を，[選択肢] の中から選び数字で答えてください。段階的に考えましょう。”

Heuristic（計 17 文）

”あなたはプロのクイズプレイヤーです。”
”あなたの真の力を見せてください。”
”あなたの努力は素晴らしい結果をもたらします。”
”自分の仕事に誇りを持ち，最善を尽くしてください。”
”挑戦を成長の機会と捉えてください。”
”本当に正解かどうか、よく考えてください。”
”間違えると生死に関わります。”
”この問題に正解すると，多額の報酬がもらえます。”

ReInstruction(計 17 文)

”\n” (省略)
”###命令 \n[問題] に回答してください。”
”###注目 \n[問題] を読み、[選択肢] の中から答えを選んで番号を出力してください。”
”###重要 \n[問題] に対して最も適当な答えを [選択肢] の中から選び、その番号を出力してください。”

A.2 チューニングプロンプト

JCommonsenseQA

”###命令
[問題] に対する回答を、[選択肢] の中から選び数字で答えてください。その際、答えの推論過程を記述してください。
[問題] に対する回答を出してください。
あなたはクイズのプロ解答者です。
[例題] : 墨を吐くものはどれ？
[選択肢例] :
0 : タコ \n 1 : さんご \n 2 : 人 \n 3 : ワニ \n 4 : 犬 \n
[解答例] : 0
[例題] : 朝食バイキングといえば、思い浮かぶ施設はなんでしょうか？
[選択肢例] :
0 : ホテル \n 1 : 海賊船 \n 2 : コンビニ \n 3 : 保健所 \n 4 : 学校 \n
[解答例] : 0
[例題] : 家屋と敷地を合わせたものの呼称は？
[選択肢] :
0 : 屋敷 \n 1 : 天井裏 \n 2 : スタジオ \n 3 : 地下室 \n 4 : アトリエ \n
[解答例] : 0
[問題] : 熱い物の温度を下げる機械は？
[選択肢]
0 : 警音器 \n 1 : ヒーター \n 2 : エンジン \n 3 : オドメーター \n 4 : 冷却器 \n
[解答] :”

JNLI

”###重要
[例題] を参考にして、[問題] で与えられる [文 1] と [文 2] の最も適当な関係を [選択肢] の中から選び出力してください。
自分の仕事に誇りを持ち、最善を尽くしてください。
[例題] :
[例文 1] : ひとりの男性がホットドックを頬張っています。
[例文 2] : 男性が帽子をかぶったままパンを食べています。
[選択肢例] :
0 : entailment1 : contradiction2 : neutral
[解答例] :
2 : neutral
[例題] :
[例文 1] : お皿に乗ったグレープフルーツと飲みかけのジュースが置かれています。
[例文 2] : トレーに置かれた白い皿の上にカットされたグレープフルーツが乗っています。
[選択肢例] :
0 : entailment1 : contradiction2 : neutral
[解答例] :
2 : neutral
[例題] :
[例文 1] : 駐車禁止標識の前でバイクが停車しているところです。
[例文 2] : 木の柱に交通標識が取り付けられており、後ろには黒いヘルメットを被った人がバイクにまたがっています。
[選択肢例] :
0 : entailment1 : contradiction2 : neutral
[解答例] :
2 : neutral
「問題」で提示される 2 つの文章を比較し、
・含意関係、つまり 0：entailment
・矛盾関係、つまり 1：contradiction
・中立関係、つまり 2：neutral
のどれに該当するか判断し、その結果を出力してください。また、その結果に至った論理的な推論過程も明記してください。
[問題] :
[文 1] : テニスコートでテニスラケットを持った人が立っている。
[文 2] : ラケットを持った人がテニスコートにいます。
[選択肢] :
0 : entailment1 : contradiction2 : neutral
[解答] :”

A.3 突然変異「言い換え」のプロンプト

モデルは”gpt-4-0613”を用いた.

突然変異プロンプト

”以下の日本語文を，同じ意味のまま言い換えてください。出力する際は言い換えた文のみを出力してください。
[日本語文] : #言い換えたいプロンプト
[言い換え] : ”

A.4 実験詳細

表 2 3 回試行後の平均正答率

Model	JCSQA			JNLI		
	学生 1	学生 2	学生 3	学生 4	学生 5	学生 6
3B	27.67	29.34	26.96	39.43	37.09	35.18
7B	84.27	84.27	80.19	36.39	40.07	34.98