

# 創造的な文生成タスクに対する LLM プロンプトの自動生成

鈴木就斗<sup>1</sup> 狩野 芳伸<sup>2</sup>

<sup>1</sup> 株式会社エイチームライフデザイン <sup>2</sup> 静岡大学

suzuki.shuto@a-tm.co.jp kano@kanolab.net

## 概要

大規模言語モデル (LLM) の性能を引き出すには、与えるプロンプトの良し悪しが重要である。プロンプト自動生成の研究もあるが、多くは自動評価しやすい唯一解のあるタスクが対象である。我々は、唯一解の無い創造的な文生成タスクに対してプロンプトの自動生成を試みた。そのための自動評価指標には埋め込み表現を用いて、期待される正解への近さと、創造性すなわち入力からの距離とを総合した指標により両者が反映されるようにした。インターネット記事作成における読者ニーズおよび記事見出しの生成を題材に、プロンプトを生成しない Few-shot Prompting と比較し、主観評価により提案手法がより有用な結果を生成できることを示した。

## 1 はじめに

近年、大規模言語モデル (LLM) の発展により、様々なタスクが高い精度で解けるようになった。しかし LLM の性能は、与えるプロンプトの品質に大きく左右される。よいプロンプトを作成するためには、解きたいタスクに対する知識や経験を多く求められることが多い。

LLM に与えるプロンプトの作成の負担を軽減するために、Auto Prompt Engineer (APE) [1] のようなプロンプト自体を LLM で自動生成する手法も提案されている。それらの手法の多く [2] は、自動評価しやすい唯一解のあるタスクを対象にしている。

一方で創造的な文生成タスク、たとえばインターネット上のメディアに掲載するための記事コンテンツ (以下、単に記事とする) の作成においては「記事の読者のニーズが何かを考える」「読者の目を引くような見出しを作成する」といった、唯一解の無い創造的なタスクが多い。こうした記事を作成するには多くの人的コストがかかるため、記事の作成のために言語モデルを利用した自動化技術による負担軽減や品質の一律化が期待されている。しかしこの

ようなタスクは正解データが与えられていたとしても生成結果を自動評価することは難しく、従来の方法でプロンプトを自動生成するには工夫が必要となる。

我々は埋め込み表現を用いることで、こうした唯一解の無い創造的な文生成タスクの自動評価ができるのではないかと考えた。埋め込み表現のベクトルは文章の意味の類似度を測る手段として用いられることが多い。予め与えられた正解データに対して、埋め込み表現間の類似度を計算することで、生成結果がどれくらい正解に近いかを自動評価できる。また与えた入力と生成結果との類似度を計算することで、生成結果がどれくらい入力から離れているかも計算できる。これによって、入力から直接連想しづらい創造的な結果が得られたかを自動評価できると考えた。

我々は、先行研究として PromptAgent[3] の手法を参考に、タスクに対して少数の入力と出力のペアを用いてプロンプトを最適化した。最適化されたプロンプトを用いて新たに生成された出力を、埋め込み表現によって評価した。これにより、唯一解の無い創造的な文生成タスクにおいてプロンプトの自動最適化を実現した。

## 2 関連研究

### 2.1 プロンプトの自動生成

プロンプトの自動生成、すなわち Auto Prompt Engineer (APE) [1] は、少数の入力と出力のペアからプロンプトを自動生成する手法である。この研究では OpenAI GPT-3[4] を用いて複数のプロンプト候補を生成し、答えが出力される対数確率を用いて生成結果を自動評価し、評価結果の高いプロンプトを採択するサイクルを複数回続けることでプロンプトを最適化している。APE で用いられた GPT-3 では日本語における性能が不十分であり、同様の手法で日本語のタスクに対して高品質な出力を得ることは

難しい。また対数確率が取得できない API ベースのクローズドなモデルに対して、APE と同様の手法を適用することは難しいという問題がある。

GPT-4[5] を利用したプロンプト自動生成の手法として PromptAgent[3] がある。PromptAgent は、入力と出力のペアと初期の単純なプロンプトを用意して、初期のプロンプトによる生成結果を LLM でエラーフィードバックすることによってプロンプトを改善する手法である。改善されたプロンプトを自動評価するために、150 件の入力と出力のペアを用意して報酬関数を計算している。その自動評価されたスコアに基づき、モンテカルロ木探索を用いて最適なプロンプトを探索している。

## 2.2 自動評価

唯一解の無い創造的な文生成タスクの自動評価としては、埋め込み表現を用いる方法が考えられる。たとえば Multilingual E5[6] などを用いて出力結果の文埋め込み表現を獲得し、正解データとの類似度を測ることで自動評価ができる。

また、ChatGPT を用いた自動評価の手法 [7] が提案されている。文章の要約や物語の生成といった創造的なタスクの評価に ChatGPT を使い、人間による手動評価との相関性が高いと報告されている。

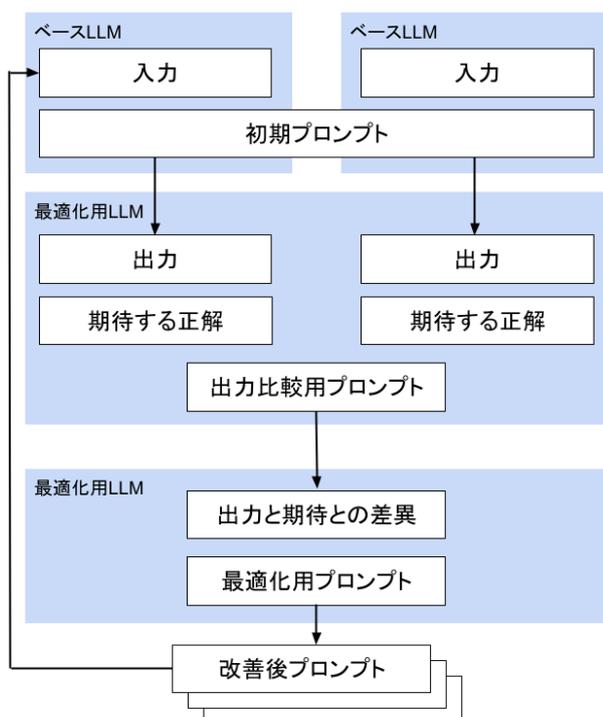


図 1 提案手法の流れ

## 3 提案手法

関連研究に挙げた PromptAgent では唯一解のあるタスクが対象であるが、唯一解のある大規模データセットではなく、少数の入力と出力のペアによってプロンプトを自動で最適化を行う手法を提案する。

具体的には、プロンプトと入力文字列から出力を生成するベース LLM と、出力と期待する正解を比較して新しいプロンプトを生成する最適化用 LLM を用いて、プロンプトを自動で最適化する。最適化の指標として、生成結果と期待する正解との近さを表す指標 A と、より創造的すなわち入力との遠さを表す指標 B を定義し、それらを総合した仕様により創造的かつ正解といえる出力を得ることを目指す。

### 3.1 プロンプトの生成

準備として、入力とそれに対して期待される正解となる出力のペアを 2 組<sup>1)</sup>と、簡易な初期プロンプトを用意する。まず入力と初期プロンプトを用いて、ベース LLM で出力を得る。次に出力と期待される正解との違いを、最適化用 LLM によって説明させる。得られた出力と正解との差異の情報をもとに最適化された新しいプロンプトを複数生成するよう、最適化用 LLM に指示を与えることでプロンプトを再度生成する (図 1)。

ベース LLM に与える、入力から生成結果を得るプロンプトは付録 A.1 に示す。

最適化用 LLM に与えるための、ベース LLM による生成結果と期待する出力の違いを説明する出力比較用プロンプトは次のようになる。

出力比較用プロンプト

以下はあるプログラム A と B に同じ入力を与えた時の出力です。  
A と B の中身を推測し、A と比べて B にはどのような特徴があるかを出力してください。

```
## 入力 1
{input}
### A
{output}
### B
{answer}

## 入力 2
```

1) 2 組とした理由は、できるだけ少ない数のペアを用いたいと考えたが、1 組ではタスクを抽象化できず出力を直接指示するようなプロンプトが生成され意図した結果とならなかったため、必要最低限の数として 2 組が必要と判断した。

```
{input}
### A
{output}
### B
{answer}
```

B の特徴:

input, answer には, 予め用意した2組の入力と期待する出力を記述する. output には, ベース LLM で生成した結果を記述する.

最適化用 LLM に与えるための, 新しいプロンプトを複数生成する最適化用プロンプトは次のようになる.

最適化用プロンプト

言語モデルのプロンプトを作成しています。正解が得られるよう現在のプロンプトを改善してください。

## 現在のプロンプト

```
{prompt}
```

## 現在のプロンプトの問題点

現在のプロンプトによる出力を A とします。得たい正解となる出力を B とします。

```
{feedback_output}
```

A と B の違いを踏まえ、B が出力されるように現在のプロンプトを改善した、`{generate_number}` 個の新しいプロンプトを出力してください。プロンプトには決して入力と出力を含めないでください。新しいプロンプトは `<START>` と `<END>` で囲んでください。

新しいプロンプト:

feedback\_output には, 先ほどの出力比較用プロンプトの出力を記述する. generate\_number には, 新しいプロンプトを生成する個数を記述する.

以上によって得られた結果から `<START>` と `<END>` で囲まれた部分を抽出したものを, 改善された新しいプロンプトとする. 改善された新しいプロンプトをベース LLM に与えることで, 新しい生成結果を得る.

### 3.2 自動評価のための指標

自動評価により生成された複数の結果のうちひとつを採用するために, 埋め込み表現を用いた. 入力  $i$ , 期待する出力  $a$ , 出発点となる簡易的なプロ

ンプトによる出力を  $o_0$ , 最適化後の新しいプロンプトによる出力を  $o_1$  とする. それぞれの文章から文ベクトルを得て, ふたつのベクトルのコサイン類似度をとったものを  $s(*,*)$  とする.

改善後の出力がどれだけ正解に近づいたかを示す指標として,  $s(a, o_1) - s(a, o_0)$  を評価指標 A とする. また改善後の出力が, どれだけ入力から離れたかを示す指標として,  $s(i, o_0) - s(i, o_1)$  を評価指標 B とする. 総合評価を  $A + B$  として, この値が最も高いプロンプトを生成結果として採用した.<sup>2)</sup>

## 4 実験

### 4.1 セットアップ

先行研究である PromptAgent の手法にならない, ベース LLM には OpenAI GPT 3.5 Turbo Nov 6 version としたが, GPT-4 などの他のモデルに置き換えも可能と考えられる. 最適化用 LLM には高い推論能力が求められるため, PromptAgent 同様に OpenAI GPT-4 Turbo Nov 6 version を用いた.

実験用データとして, インターネット記事作成を題材に, その読者ニーズ生成と目次見出し生成 (付録 B) を想定したタスクを合計 30 件用意した. 内訳は, キーワードを入力とし, 潜在ニーズを生成するタスク 10 件, 文を入力とし, 見出しを生成するタスク 10 件, キーワードを入力とし, 見出しを生成するタスク 10 件となっている.

評価にあたって必要になる埋め込み表現は, OpenAI text-embedding-ada-002 を用いて生成した.

1 回の最適化において 3 個の新しいプロンプトを生成するよう指示した. なおタスクによっては 3 個生成するよう指示しても期待通りに 3 個生成されないことがあるが, 生成された分だけ評価をした. 生成されたプロンプトから改善された出力を得て, 評価指標 A と B を計算し, 評価指標 A と B の和が最も高くなるプロンプトを採用した.

### 4.2 ベースライン

提案手法に対するベースラインとして Few-shot Learning あるいは Few-shot Prompting と呼ばれる手法と比較した. Few-shot Prompting は, 少数の入力と出力のペアを用いる必要がある点で提案手法と共通しているが, プロンプトの自動生成は行わず入出

2) 厳密には演算前に A と B を互いの分布を合わせる補正が必要と考えられるが, 本研究においては A と B の分布がほとんど同じであったため取得した値をそのまま使った.

表 1 実験結果

	提案手法	ベースライン
評価指標 A	0.00769	0.01995
評価指標 B	0.01431	0.00053
評価指標 A+B	0.02200	0.02048

力の例を用いて新しい出力を直接得る手法である。Few-shot Prompting として、以下のプロンプトを用いた。

ベースラインとして用いたプロンプト

```
input: {input_1}
output: {output_1}
input: {input_2}
output:
```

## 5 評価結果

### 5.1 自動評価

自動評価として、評価指標 A と B、および A+B の値の平均値を算出した (表 1)。評価指標 A+B において、提案手法がベースラインよりも高い値を得ていることが分かる。

### 5.2 主観評価

人手での主観評価として、生成結果の文章を読んだ上で、どちらの出力が有用かを 1 名に選択させたところ、約 62 % のタスクで提案手法の方が有用であると判断した。

次に、提案手法で最適化する際に自動評価として評価指標 A のみを用いる場合と、A+B を用いた場合についての変化を確認した。A のみを用いた場合と A+B を用いた場合の出力は一致する事があるが、異なる出力をしたケースについてどちらの出力が有用かを 1 名に選択させたところ、約 64 % のタスクで A+B を用いた場合の方が有用であると判断した。

## 6 議論と分析

まず、評価指標について議論する。期待する正解への近さを表す評価指標 A はベースラインの方が高くなった。Few-shot Prompting は与えられた正解を直接用いて新しい生成結果を得る手法のため、より与えられた正解に忠実な出力を得られやすい傾向があると考えられる。一方で、入力からの距離を表す評価指標 B は提案手法の方が高くなった。提案手法の方が入力から連想しづらい出力を得やすくなるた

め、唯一の正解が無い創造的なタスクに向いているのではないかと考える。

次に、実際の生成結果を分析する。「プログラミング講座」という入力に対して、記事読者のニーズを出力させた結果を次に示す。なお期待する正解は「プログラミングに興味はないが、キャリアアップしたい」とした。もう一組の例として与える入出力のペアは入力「30代転職」に対して「転職するつもりは無いが、給料を上げたい」とした。

ベースライン手法による生成結果

プログラミングの講座を受けてスキルを向上させたい。

提案手法による生成結果

読者はプログラミング講座を受講することで、プログラミングスキルを向上させたいという表面的な要求の背後に、将来のキャリアアップや新しい仕事の機会を得るための深層のニーズや目的があると考えられます。

いずれの手法も「プログラミング講座」という入力から「スキルを向上させたい」という意味の出力を得ている。提案手法においてはさらにキャリアアップや新しい仕事の機会といった、入力から直接連想しづらい創造的な出力を得た。

提案手法において改善されたプロンプトは次のようになった。

提案手法により改善されたプロンプト

提示されたキーワードから読者の表面的な要求を超えた深層のニーズや目的を探り、それらを明らかにする分析を行ってください。

このように入力と出力のペアから求められる情報を抽象化し、より有用な出力が得られるようプロンプトを最適化できた。

## 7 おわりに

唯一解の無い創造的な文生成タスクのためにプロンプトを自動生成する手法を提案し、より有用性の高い結果が得られることを示した。今後はプロンプトの最適化を繰り返し実行することで、さらに有用な出力を得ることを目指したい。その場合、LLM による評価と生成を繰り返し実行する必要があるため、より軽量なモデルで実行したいと考えている。

## 参考文献

- [1] 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 **Proceedings of the Eleventh International Conference on Learning Representations (ICLR)**, 2022.
- [2] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with “gradient descent” and beam search. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, 2023.
- [3] Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization, October 2023. [arXiv:2310.16427](https://arxiv.org/abs/2310.16427).
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.
- [5] Josh Achiam et al. Gpt-4 technical report, 2023. [arXiv:2303.08774](https://arxiv.org/abs/2303.08774).
- [6] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, December 2022. [arXiv:2212.03533](https://arxiv.org/abs/2212.03533).
- [7] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study. In Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini, editors, **Proceedings of the 4th New Frontiers in Summarization Workshop**, pp. 1–11, Hybrid, December 2023. Association for Computational Linguistics.

## A プロンプトテンプレート

### A.1 入力から出力を得るプロンプト

ベース LLM で出力を得るプロンプトは以下。

入力から出力を得る

```
{prompt}

## 入力
{input_text}

## 出力の形式
{answer_format}

出力:
```

prompt には、最適化対象となる初期プロンプトを記述する。input\_text には入力を記述する。answer\_format には「答えを一文で出力して」といった出力の形式を簡単に記述する。

## B 実際の生成例

### B.1 読者のニーズを考える例

入力「30代転職」に対して、期待する正解を「転職するつもりは無いが、給料を上げたい」とした例を次に示す。

初期のプロンプト

次の入力について読者のニーズを教えてください

初期のプロンプトによる生成結果

30代の読者が転職に関する情報を求めている。

LLMにより改善された新しいプロンプト

読者が提供したキーワードを超えて、その背後にある潜在的な動機や欲求を分析し、それに基づいた情報やアドバイスを提供してください。

新しいプロンプトによる生成結果

転職を考える30代の方は、自己成長やキャリアアップの欲求が強い可能性があります。新しい環境での挑戦やスキルの磨き直しを通じて、自分自身を成長させることを意識してみてください。

また、ベースラインとした Few-shot Prompting による生成結果を次に示す。

Few-shot Prompting による生成結果

転職を考えているが、年齢的に不安がある

改善されたプロンプトにより、「自己成長」や「挑戦」といった要素が得られた。

### B.2 記事の見出しを作成する例

入力「初心者プログラミング始める」に対して、期待する正解を「初心者でもプログラミングは始められる？」とした例を次に示す。

初期のプロンプト

入力について、記事の目次の一項目となる見出しを作成して

初期のプロンプトによる生成結果

プログラミングの初心者が始めるための入力方法

LLMにより改善された新しいプロンプト

記事の目次に含めるため、入力されたキーワードを使って疑問形で読者の興味を引く見出しを生成してください。

新しいプロンプトによる生成結果

初心者でもプログラミングを始めるべき？

また、ベースラインとした Few-shot Prompting による生成結果を次に示す。

Few-shot Prompting による生成結果

初心者でもプログラミングを始めることはできますか？

この例においては、提案手法とベースラインとで大きな違いは見られなかった。