

プロンプト手法と現在の各生成AIでの効果 (2025年5月版)

はじめに

生成AIの進化は目覚ましく、その能力を最大限に引き出す「プロンプトエンジニアリング」の重要性はますます高まっています。しかし、技術の進歩とともに、かつて有効とされたプロンプト術が必ずしも最善ではなくなったり、特定のAIモデルでは異なるアプローチが求められたりするケースも散見されるようになりました。

本書は、生成AI活用のスペシャリストを目指す方々、あるいは日々の業務でAIをより効果的に使いたいと考えているすべての方々に向けて、主要なプロンプト手法を整理し、現在の代表的な生成AIモデルにおける効果を比較検証することを目的としています。

過去から現在までに名前が付けられてきたプロンプト手法を概観し、それぞれのメリット・デメリット、さらには逆効果になり得るパターンまで深掘りします。そして、2025年5月末時点で注目される主要な生成AI（OpenAI GPTシリーズ、Microsoft 365 Copilot、Anthropic Claudeシリーズ、Google Geminiシリーズ）に対して、各手法がどの程度の効果をもたらすのかを評価し、その理由を解説します。

この一冊を通じて、読者の皆様が各生成AIの特性を理解し、タスクに応じた最適なプロンプトを選択・設計するための一助となれば幸いです。

第1章: プロンプトエンジニアリングの潮流

生成AIの黎明期には、AIに単純な指示を与えるだけで、ある程度の成果が得られました。しかし、モデルが高度化・複雑化するにつれて、その潜在能力を十分に引き出すためには、より洗練されたコミュニケーション、すなわち質の高いプロンプトが不可欠となりました。

初期のプロンプトは、質問応答や文章生成といった直接的なタスク指示が中心でした。その後、AIにより明確な文脈や期待する出力形式を与える「Few-Shotプロンプティング」や「テンプレートベース」の手法が登場し、出力の質と安定性が向上しました。

近年では、AIに思考プロセスをステップバイステップで記述させる「Chain of Thought (CoT)」や、外部ツールとの連携を可能にする「ReAct」など、より高度な推論や問題解決能力を引き出す手法が注目されています。これらの手法は、AIの「ブラックボックス」的な部分を少しでも可視化し、制御しようとする試みとも言えます。

しかし、AIモデルの進化はプロンプト手法の有効性にも影響を与えます。例えば、非常に高度な文脈理解能力を持つ最新モデルに対しては、過度に詳細なFew-Shotの例が逆に創造性を制限したり、冗長な指示となったりする可能性も指摘されています。また、特定のタスクにおいては、CoTのような思考プロセスを強制することが、かえって非効率的であったり、誤った推論

を誘発したりするリスクも認識され始めています。

さらに、生成AIの種類によっても、得意とするタスクやプロンプトへの応答性が異なります。汎用性の高いモデルもあれば、特定の用途にチューニングされたモデルも存在します。これらの特性を理解せずに画一的なプロンプトを使用しているだけでは、期待する成果を得ることは難しいでしょう。

このような背景から、今一度、既存のプロンプト手法を棚卸しし、最新のAIモデル群との相性を再評価する必要性が高まっているのです。

第2章: 基本的なプロンプト手法とその応用

ここでは、プロンプトエンジニアリングの基礎となる代表的な手法について、その概要、有効なケース、逆効果になるパターン、そして具体的なサンプルプロンプトを紹介します。

2.1 Zero-Shot Prompting (ゼロショット・プロンプティング)

- 概要: AIモデルに事前の例(ショット)を与えずに、タスクの指示のみで応答させる手法です。モデルが持つ広範な事前知識と汎化能力に依存します。
- 有効なケースとメリット:
 - 単純な質問応答、一般的な文章生成、要約、翻訳など、AIが事前学習で十分に知識を獲得していると考えられるタスク。
 - 手軽で迅速に試せるため、アイデアの初期検証や簡単なタスクに適しています。
 - プロンプト作成のコストが低い。
- 逆効果になるパターンや注意点:
 - 複雑な推論が必要なタスク: 複数のステップや論理的な繋がりを理解する必要がある問題では、指示だけでは不十分なことが多いです。
 - 専門知識が深く必要なタスク: ニッチな分野や最新の専門知識を要する場合、AIが誤った情報や古い情報を生成する可能性があります。
 - 指示が曖昧な場合: 期待する出力形式や内容が不明確だと、AIが意図しない回答を生成しやすくなります。
 - 創造的だが制約の多いタスク: 例えば「特定のキャラクターになりきって、特定のテーマで詩を書いてください」といった場合、指示だけではキャラクターやテーマのニュアンスを掴みきれないことがあります。
- サンプルプロンプト:
 - (翻訳) フランス語に翻訳してください: こんにちは、世界。
 - (要約) 以下の文章を3行で要約してください: [長文テキスト]
 - (質問応答) 日本の首都はどこですか？

2.2 Few-Shot Prompting (フューショット・プロンプティング) / One-Shot Prompting (ワンショット・プロンプティング)

- 概要: AIに応答の前に、いくつかの例(ショット)とその期待される出力形式や内容を提示する手法です。One-Shotは1つの例、Few-Shotは複数の例を与えます。これにより、AIはタスクのパターンや文脈をより具体的に理解し、期待に近い出力を生成しやすくなります。
- 有効なケースとメリット:
 - Zero-Shotでは上手くいかない、少し複雑なタスクや特定の出力形式が求められる場合。
 - 新しい概念や特定のスタイルをAIに学習させたい場合(In-Context Learning)。
 - 出力の一貫性を高めたい場合。
- 逆効果になるパターンや注意点:
 - 例の質が低い、または偏っている場合: 提示された例に誤りがあったり、特定の傾向に偏っていたりすると、AIもそれに倣った不正確または偏った出力を生成する可能性があります。
 - 例がタスクの本質からずれている場合: 例がタスクの核心的な部分を捉えていないと、AIは誤ったパターンを学習してしまいます。
 - 例の数が不適切(少なすぎる/多すぎる)な場合: 少なすぎると効果が薄く、多すぎるとモデルによっては混乱したり、プロンプトが長くなりすぎて処理効率が悪くなったりすることがあります。特に最新の高性能モデルでは、過剰な例示が冗長と判断されることもあります。
 - モデルが例に過剰適合する(オーバーフィッティング): 提示された例の表面的なパターンに囚われすぎて、少し異なる入力に対して柔軟に対応できなくなることがあります。
 - 例のフォーマットが一貫していない場合: 入力と出力のペアの形式がバラバラだと、AIがパターンを認識しにくくなります。
- サンプルプロンプト (Few-Shot):

ユーザー: このリンゴは赤いです。
AI: This apple is red.

ユーザー: あの猫はとても可愛いです。
AI: That cat is very cute.

ユーザー: その本は面白いですか?
AI: Is that book interesting?

ユーザー: 私の名前は山田です。
AI:
(期待する出力: My name is Yamada.)

2.3 Role-Based Prompting (ロールベース・プロンプティング)

- 概要: AIに特定の役割(ペルソナ)を割り当てることで、その役割になりきった口調、視点、専門知識に基づいた応答を促す手法です。「あなたは〇〇の専門家です」「〇〇として振る舞ってください」といった指示が典型的です。
- 有効なケースとメリット:
 - 特定の専門分野に関するアドバイスや解説を求める場合。
 - 特定のキャラクターや文体での文章生成を期待する場合。
 - 対話型AIに特定の応対スタイル(例:フレンドリー、フォーマル)を求める場合。
 - 出力のトーンや深さをコントロールしやすくなります。
- 逆効果になるパターンや注意点:
 - 役割設定がタスクと無関係または不適切な場合: 例えば、技術的な質問に対して「詩人として教えてください」と指示しても、有用な回答は得られにくいでしょう。
 - 過度に複雑または曖昧な役割設定: AIが役割を理解できず、中途半端な応答になることがあります。
 - 役割に固執しすぎる: 設定された役割に囚われすぎて、必要な情報やより客観的な視点を提供できなくなることがあります。
 - 倫理的に問題のある役割設定: 差別的、攻撃的、または不適切なペルソナを設定すると、AIもそれに沿った好ましくない出力をする可能性があります。
 - モデルの能力を超える役割: AIが持っていない専門知識や経験を要求する役割(例:「未来を予知する占い師」)は、効果が限定的です。
- サンプルプロンプト:
 - あなたは経験豊富なマーケティングコンサルタントです。中小企業がSNSを活用して顧客を増やすための具体的な戦略を3つ提案してください。
 - 江戸時代の商人になりきって、現代のスマートフォンについてどう思うか語ってください。

2.4 Template-Based Prompting (テンプレートベース・プロンプティング)

- 概要: 事前に定義された構造(テンプレート)に、可変の情報を埋め込む形でプロンプトを作成する手法です。出力形式を厳密に制御したい場合や、定型的なタスクを効率的に処理したい場合に有効です。
- 有効なケースとメリット:
 - 報告書の自動生成、メールの作成、データの整形など、出力形式が定まっているタスク。
 - 一貫性のある出力を大量に生成する必要がある場合。
 - ユーザーが入力する情報が明確で、それを元にAIが補完や加工を行う場合。
 - プロンプトの再利用性が高い。
- 逆効果になるパターンや注意点:
 - 創造性や独自性が求められるタスク: 厳格なテンプレートは、AIの自由な発想を妨げ、画一的な出力になりがちです。
 - テンプレートの設計が不適切な場合: 必要な情報が欠落していたり、逆に不要な制約

が多すぎたりすると、期待する品質の出力が得られません。

- 柔軟性に欠ける: 想定外の入力や状況の変化に対応しにくいことがあります。
- テンプレートが複雑すぎる場合: 管理が煩雑になったり、AIがテンプレートの意図を正確に解釈できなかったりすることがあります。

- サンプルプロンプト:

件名:【会議議事録】[会議名]

日時:[YYYY年MM月DD日 HH:MM~HH:MM]

場所:[場所]

出席者:[出席者リスト]

議題:

1. [議題1]

2. [議題2]

決定事項:

- [AIに記述させる部分]

今後のアクション:

- [担当者]:[タスク](期限:[YYYY年MM月DD日])

- [AIに記述させる部分]

第3章: 高度なプロンプト手法とその応用

基本的な手法に加え、AIの推論能力や問題解決能力をより引き出すための高度な手法も開発されています。

3.1 Chain of Thought (CoT) Prompting (思考の連鎖プロンプティング)

- 概要: AIに最終的な答えを出す前に、その答えに至るまでの中間的な思考プロセスや推論のステップを明示的に記述させる手法です。これにより、特に複雑な推論問題において、AIがより正確な答えを導き出しやすくなります。「ステップバイステップで考えてください」といった指示や、Few-Shotで思考プロセスを含めた例を示すことで誘発します。
- 有効なケースとメリット:
 - 算術問題、論理パズル、常識的推論など、段階的な思考が必要なタスク。
 - AIの思考プロセスを可視化し、なぜその結論に至ったのかを理解するのに役立ちます。
 - 複雑な問題に対する正解率が向上することが多くの研究で示されています。
- 逆効果になるパターンや注意点:

- 単純な情報検索タスク: 思考プロセスが不要なタスクでは、冗長な出力となり、効率が悪くなります。
- 創造性が重視されるタスク: 論理的な思考ステップを強制することが、自由な発想や直感的なアイデア出しを妨げる場合があります。
- 間違った思考プロセスを提示した場合 (**Few-Shot CoT**): 例として提示した思考プロセスに誤りがあると、AIもその誤ったプロセスに倣ってしまい、結果的に間違った答えを導く可能性があります。
- AIが「もっともらしい嘘」の思考プロセスを生成する可能性: 必ずしも正しい推論ではなく、それっぽく見える思考ステップを生成してしまうことがあります。
- 計算コストの増加: 思考プロセスを記述する分、生成するトークン数が増え、応答時間やコストが増加する傾向があります。
- 最新モデルでは自律的に内部で**CoT**類似の処理を行っている場合: 最新の高性能モデルでは、明示的なCoT指示がなくとも、内部的に複雑な推論を行えるため、CoTプロンプトが必ずしも大きな改善をもたらさないか、冗長になることがあります。
- サンプルプロンプト (**Zero-Shot CoT**):
問題:
カフェに最初に5人のお客さんがいました。その後、3人のお客さんが来て、2人のお客さんが出ていきました。今、カフェには何人のお客さんがいますか？
ステップバイステップで考えて答えを導き出してください。

3.2 ReAct (Reasoning and Acting) Prompting (推論と行動プロンプティング)

- 概要: AIが推論(Reasoning)と行動(Acting)を交互に行うことで、より複雑なタスクを解決するフレームワークです。AIはまずタスクを達成するための計画を立て、次に必要な情報を得るための行動(例: 外部ツールを使った検索、計算)を決定し、その結果を観測(Observation)して次の推論に繋がります。これにより、AIは自身の知識だけでは解決できない問題に対して、外部リソースを活用しながら答えを導き出すことができます。
- 有効なケースとメリット:
 - 最新情報や専門的なデータベースへのアクセスが必要な質問応答。
 - 複数の情報源を組み合わせて答えを生成する必要があるタスク。
 - 複雑な計算やデータ処理を伴う問題解決。
 - AIの行動プロセスが透明化され、デバッグや改善がしやすくなります。
- 逆効果になるパターンや注意点:
 - ツール連携が不要な単純タスク: ReActの枠組み自体がオーバーヘッドとなり、効率が悪くなります。
 - 利用可能なツールが限定的または不適切な場合: 必要な情報を取得できるツールがない、あるいはツールの性能が低い場合、ReActの効果は発揮されません。
 - ツールのAPI制限やエラーハンドリングが困難な場合: 外部ツールとの連携には、API呼び出し回数の制限や予期せぬエラーへの対応が求められ、実装の複雑性が増

します。

- 思考と行動のステップが過剰に多くなる場合: タスク解決までに多くのステップが必要になると、遅延や途中でエラーが発生するリスクが高まります。
- AIが適切なツール選択や行動計画を立てられない場合: 特に複雑なタスクでは、AIが最適な行動シーケンスを設計できないことがあります。
- サンプルプロンプト (概念的な例):
質問: 今日の東京の天気と、明日の大阪の天気を教えてください。

思考: ユーザーは東京の今日の天気と大阪の明日の天気を知りたい。それぞれの情報を取得するために天気検索ツールを使う必要がある。

行動: SearchWeatherTool("東京 今日の天気")

観察: [東京の天気情報が返される]

思考: 東京の天気は取得できた。次に大阪の明日の天気を調べる。

行動: SearchWeatherTool("大阪 明日の天気")

観察: [大阪の天気情報が返される]

思考: 両方の情報を取得できたので、まとめて回答する。

回答: 今日の東京の天気は[天気]です。明日の大阪の天気は[天気]です。

(実際には、ReActはエージェントが内部的にこのような思考と行動のループを実行するフレームワークであり、ユーザーが直接全てのステップを指示するわけではありません。)

第4章: 生成AIモデル別 プロンプト手法適合性評価 (2025年5月時点)

ここでは、主要な生成AIモデルと、これまで解説してきたプロンプト手法との相性を評価します。

4.1 対象とする生成AIモデルについて

本評価では、2025年5月末時点で代表的ないくつかの大規模言語モデル(LLM)を取り上げます。

- OpenAI GPTシリーズ:
 - **GPT-3.5 (主に gpt-3.5-turbo):** 広く普及しており、多くのアプリケーションの基盤となっています。高速かつ比較的低コストですが、最新モデルと比較すると推論能力や指示追従性で見劣りする場合があります。
 - **GPT-4 (主に gpt-4-turbo など):** GPT-3.5から大幅に性能が向上し、より複雑な指示の理解、高度な推論、創造性の点で優れています。
 - **GPT-4o (gpt-4o):** GPT-4 Turboの能力を維持しつつ、速度と効率を大幅に改善し、マルチモーダル対応を強化した最新モデル(本評価執筆時点)。テキスト処理においても高い性能を発揮します。
- Microsoft 365 Copilot:
 - Microsoftの各種サービス(Word, Excel, PowerPoint, Outlook, Teamsなど)に統合

されたAIアシスタント。バックエンドにはOpenAIのGPTモデル(主にGPT-4やそれ以降の最新モデル)が使用されていますが、Microsoftによる独自のチューニングやデータ連携、セキュリティ対策が施されています。そのため、同じGPTモデルを直接利用する場合とは挙動や得意分野が異なることがあります。本評価では、最新のMicrosoft 365 Copilotの一般的な傾向を対象とします。

- **Anthropic Claude**シリーズ:
 - **Claude 3 Opus:** Anthropic社の最も高性能なモデル(本評価執筆時点)。特に長いコンテキストの処理能力、複雑な指示への追従性、そして「Constitutional AI」による安全性と倫理性に配慮した応答生成に強みがあります。
- **Google Gemini**シリーズ:
 - **Gemini 1.5 Pro:** Googleの高性能マルチモーダルモデル。長大なコンテキストウィンドウ(最大100万トークン)と高度な推論能力が特徴です。
 - **Gemini 1.5 Flash:** Gemini 1.5 Proの能力を維持しつつ、速度と効率を重視した軽量モデル。大規模なバッチ処理や迅速な応答が求められるタスクに適しています。

注記: ユーザー様ご指定の「Chat GPT 4.1」「ChatGPT o4」「Claude 4.0」「Gemini 2.5 Flash」「Gemini 2.5 Pro」というモデル名は、2025年5月末時点で公式にリリースされている名称とは異なります。本評価では、それぞれ現行の最新世代・高性能モデルであるGPT-4/GPT-4o、Claude 3 Opus、Gemini 1.5 Pro/Flashを対象として解説いたします。

4.2 プロンプト手法 効果比較表

以下の表は、各プロンプト手法がそれぞれのAIモデルに対してどの程度の効果をもたらすかの一般的な目安を示したものです。

評価基準:

- ◎: 非常に効果的で、多くの場合に推奨される
- ○: 効果的であり、一般的に良い結果をもたらす
- △: 効果がある場合もあるが、状況やタスクに依存する。あるいは、他の手法の方が優れていることが多い
- ×: あまり効果がない、または逆効果になる可能性がある。注意が必要

| プロンプト 手法 | GPT-3.5 | GPT-4 | GPT-4o | Microso ft 365 Copilot | Claude 3 Opus | Gemini 1.5 Pro | Gemini 1.5 Flash |
|-------------------|---------|-------|--------|------------------------------|------------------|-------------------|------------------------|
| Zero-Sh ot | △ | ○ | ◎ | ○ | ◎ | ◎ | ○ |
| Few-Sh ot/One- | ○ | ◎ | ○ | ○ | ○ | ○ | ○ |

| Shot | | | | | | | |
|------------------------|---|---|---|---|---|---|---|
| Role-Based | ○ | ◎ | ◎ | ◎ | ◎ | ◎ | ◎ |
| Template-Based | ◎ | ○ | ○ | ◎ | ○ | ○ | ○ |
| Chain of Thought (CoT) | △ | ○ | ○ | △ | ○ | ◎ | ○ |
| ReAct | × | △ | ○ | △ | △ | ○ | △ |

※この表はあくまで一般的な傾向を示すものであり、具体的なタスク、プロンプトの質、各AIモデルのマイナーアップデートによって効果は変動します。

4.3 効果比較表の詳細解説

手法ごとの考察

- **Zero-Shot Prompting:**

- **GPT-3.5 (△):** 単純なタスク以外では指示の解釈が不十分だったり、期待する深さの回答が得られなかったりすることが多いです。明確な指示と文脈設定がより重要になります。
- **GPT-4, GPT-4o, Claude 3 Opus, Gemini 1.5 Pro (◎～○):** これらの高性能モデルは、Zero-Shotでも非常に高い能力を発揮します。特にGPT-4o、Claude 3 Opus、Gemini 1.5 Proは、複雑な指示やニュアンスの理解力に優れており、質の高いZero-Shotプロンプトで多くのタスクをこなせます。GPT-4も依然として強力です。
- **Microsoft 365 Copilot (○):** バックエンドのモデル性能は高いものの、アプリケーション統合や安全性への配慮から、やや保守的な応答になることがあります。明確な指示が有効です。
- **Gemini 1.5 Flash (○):** 高速性を重視しつつも、Zero-Shotでの応答品質は良好です。Pro版ほどの深みは出にくいかもしれませんが、多くの一般的なタスクで十分な性能を示します。
- **昔との変化:** 初期モデルほどZero-Shotの限界は低かったですが、最新モデルではその適用範囲が格段に広がりました。「とりあえずZero-Shotで試す」価値は非常に高まっています。

- **Few-Shot/One-Shot Prompting:**

- **GPT-3.5 (O):** Zero-Shotでは不足しがちな文脈や出力形式を補うのに有効です。
- **GPT-4, GPT-4o, Claude 3 Opus, Gemini 1.5 Pro/Flash (O~◎):** これらのモデルはFew-Shotから効率よく学習しますが、過剰な例示は逆に冗長と見なされたり、モデルの柔軟性を損ねたりする可能性も出てきました。特にGPT-4oやClaude 3 OpusのようにZero-Shot性能が高いモデルでは、Few-Shotの必要性が相対的に低下するタスクもあります。例の質と量がより重要になります。「少ないが高品質な例」が鍵です。
- **Microsoft 365 Copilot (O):** 特定のフォーマットやスタイルを指示する際に有効です。
- **昔との変化:** かつてはFew-Shotが性能向上の常套手段でしたが、最新モデルでは「必ずしもFew-Shotが良いとは限らない」ケースが増えています。Zero-Shotで試行し、それで不十分な場合に質の高いFew-Shotを検討するという流れがより一般的になっています。
- **Role-Based Prompting:**
 - **全般 (◎~O):** ほとんどの最新モデルで非常に効果的です。役割を与えることで、出力のトーン、専門性、視点をコントロールしやすくなります。モデルの指示追従性が向上したことで、より複雑な役割設定にも応えやすくなっています。
 - **Microsoft 365 Copilot (◎):** アプリケーション内での特定のタスク(例: メール作成、プレゼン資料のアイデア出し)において、ユーザーの意図する役割(例: プロフェッショナル、親しみやすい同僚)をAIに与えることで、より文脈に即した支援を得やすくなります。
 - **昔との変化:** 以前のモデルでも有効でしたが、最新モデルではより自然で深い役割演技が可能になっています。ただし、不適切な役割設定によるバイアスの助長には引き続き注意が必要です。
- **Template-Based Prompting:**
 - **GPT-3.5 (◎):** 出力形式を厳密に制御したい場合に非常に有効で、安定した結果を得やすいです。
 - **GPT-4, GPT-4o, Claude 3 Opus, Gemini 1.5 Pro/Flash (O):** 依然として有効ですが、これらのモデルは柔軟性が高いため、厳格すぎるテンプレートは逆に創造性を制限する可能性があります。構造化されたデータ出力など、明確なフォーマットが必要な場合には引き続き有効です。
 - **Microsoft 365 Copilot (◎):** 定型的なドキュメント作成やデータ入力支援など、Officeアプリケーションとの連携においてはテンプレートベースのアプローチが効果を発揮しやすいです。
 - **昔との変化:** モデルの理解力向上により、ガチガチのテンプレートでなくても、自然言語による指示である程度フォーマットを制御できるようになってきました。テンプレートの「厳格さ」の塩梅が重要です。
- **Chain of Thought (CoT) Prompting:**
 - **GPT-3.5 (Δ):** CoTを誘発する能力が低く、効果は限定的です。無理にCoTをさせよ

うとすると、不自然な出力になることがあります。

- **GPT-4, GPT-4o (O):** 推論能力が高いため、CoTによって複雑な問題の正解率が向上します。ただし、モデル自身が内部的に高度な推論を行うため、Zero-ShotでもCoTに近い思考プロセスを自発的に展開することがあり、明示的なCoTが常に必要とは限りません。
 - **Claude 3 Opus (O):** 複雑な推論タスクにおいてCoTは有効です。特に長い文脈での論理的整合性を保つのに役立ちます。
 - **Gemini 1.5 Pro (◎):** 長大なコンテキスト処理能力と高度な推論能力により、CoTとの相性が非常に良いです。複雑な問題解決において、思考プロセスを明示させることで、より信頼性の高い結果を導き出すことが期待できます。
 - **Gemini 1.5 Flash (O):** Pro版ほどではないものの、CoTによって推論タスクの性能向上が期待できます。
 - **Microsoft 365 Copilot (Δ):** アプリケーション統合の文脈では、ユーザーは最終的な成果物を求めることが多く、CoTのような中間的な思考プロセスを明示的に要求するケースは少ないかもしれません。また、応答速度も重視されるため、冗長になる可能性があります。
 - 昔との変化: CoTは推論能力向上のブレイクスルーでしたが、最新モデルではその効果がタスクやモデルのベース能力に依存するようになってきました。「CoT万能」ではなく、必要な場面で見極めて使う必要があります。単純なタスクや、モデルが既に高い推論能力を持つ場合には逆効果(冗長、遅延)になるという認識が広まっています。
- **ReAct Prompting:**
 - **GPT-3.5 (×):** 外部ツールとの連携や複雑な行動計画を自律的に行う能力はほとんどありません。
 - **GPT-4 (Δ~O):** APIなどを介して外部ツールと連携させることでReAct的な動作を実装することは可能ですが、まだ限定的であり、開発側の作り込みが重要です。モデル自体が完全に自律してReActを実行するわけではありません。
 - **GPT-4o (O):** より高度な関数呼び出し機能などが強化されており、ReAct的なフレームワークを構築しやすくなっています。エージェントとしての能力が向上しています。
 - **Claude 3 Opus (Δ):** ツール連携機能はまだ発展途上であり、ReActを本格的に活用するには制約があります。
 - **Gemini 1.5 Pro (O):** GoogleのVertex AI Agent Builderなどのプラットフォームと組み合わせることで、ReActに類似した高度なエージェント機能を実現できます。モデル自体の推論能力と長文脈処理がこれを支えます。
 - **Gemini 1.5 Flash (Δ):** Pro版に比べてエージェントとしての能力は限定的ですが、ツール連携の基盤としては機能し得ます。
 - **Microsoft 365 Copilot (Δ):** CopilotはMicrosoft Graphなどを通じて内部的に情報アクセスやツール連携を行います。ユーザーがReActのような明示的な「思考と

行動の連鎖」をプロンプトで細かく指示して制御する形にはなりにくいです。Copilot自体がReAct的なエージェントとして振る舞うイメージに近いでしょう。

- 昔との変化: ReActは概念としては強力ですが、実際にエンドユーザーがプロンプトレベルでReActを意識するよりも、AIエージェントやプラグインといった形で、バックエンドでReAct的な処理が行われるのが主流になりつつあります。プロンプトで直接ReActの思考ステップを書かせるというよりは、「必要なツールを使ってこのタスクを達成して」といった高レベルな指示で、AIが内部的にReActに近い動作をするイメージです。

AIモデルごとの考察とチューニングの影響

● OpenAI GPTシリーズ (GPT-4, GPT-4o):

- 非常に汎用性が高く、ほとんどのプロンプト手法に対して良好な応答を示します。特に指示追従性、推論能力、創造性のバランスが良いです。GPT-4oはGPT-4の能力を維持しつつ効率化されているため、より多くの場面で活躍が期待されます。
- Zero-ShotやRole-Basedでのパフォーマンスが高く、Few-Shotの必要性は相対的に低下傾向にあります。CoTも有効ですが、モデルが自律的に行う推論も高度なため、常に必要とは限りません。

● Microsoft 365 Copilot:

- バックエンドはGPT-4(あるいはそれ以降のモデル)ですが、Microsoftのサービスとの統合、企業ユースケースへの最適化、セキュリティとコンプライアンスへの配慮といったチューニングが施されています。
- このチューニングにより、以下のような傾向が見られることがあります。
 - より実践的で安全な回答: 創造性や奇抜さよりも、ビジネスシーンで役立つ、誤解の少ない回答を生成しやすいです。
 - Microsoft Graphとの連携: Outlookのメール、Teamsのチャット、SharePointのファイルなど、ユーザーのMicrosoft 365内のデータに基づいた応答が可能です。これは他の汎用AIにはない大きな特徴です。
 - テンプレートベースやRole-Basedとの相性が良い: 定型的な業務支援や、特定の役割(例: 会議のファシリテーター、レポート作成者)をCopilotに与えることで、その能力を最大限に引き出しやすいです。
 - CoTやReActのような明示的な思考プロセス要求は馴染みにくい: Copilotは「賢いアシスタント」として、ユーザーに思考の過程を見せるよりも、直接的な解決策や成果物を提示する傾向があります。
- 同じGPTベースでも、Copilotでは「特定の業務ドメインにおける生産性向上」という明確な目的に沿ってチューニングされているため、汎用のChatGPTとは得意なプロンプトのニュアンスや効果的な使い方が異なる場合があります。

● Anthropic Claude 3 Opus:

- 長いコンテキストの処理能力と、複雑でニュアンスに富んだ指示への理解力に優れています。Role-Basedや、詳細な指示を含むZero-Shotで高い性能を発揮します。

- CoTも有効で、特に長文の読解や生成を伴う推論タスクで強みを見せます。
- 倫理的な配慮やハルシネーション(もっともらしい嘘)の抑制に力を入れているため、安定した応答が期待できます。
- **Google Gemini 1.5 Pro/Flash:**
 - Gemini 1.5 Proは、その驚異的な長文脈処理能力(最大100万トークン)により、大量の情報を一度に処理し、それを踏まえた深い推論が可能です。これにより、複雑なドキュメント群に基づいた要約や分析、CoT、Few-Shot(大量の例を提示可能)などで非常に高いポテンシャルを持ちます。
 - Gemini 1.5 Flashは、Proの能力を維持しつつ高速化・効率化を図っており、多くのタスクで優れたコストパフォーマンスを発揮します。Zero-ShotやRole-Basedでも十分な性能を示します。
 - マルチモーダル性が特徴ですが、テキストベースのプロンプティングにおいてもその高度な理解力と推論力は健在です。

第5章: 未来のプロンプトエンジニアリング

生成AIの進化は止まるところを知りません。モデルの能力が向上するにつれて、プロンプトエンジニアリングのあり方も変化し続けるでしょう。

- より自然言語に近い対話へ: モデルの理解力が高まることで、厳密な形式のプロンプトよりも、人間同士がコミュニケーションを取るような、より自然で文脈に依存した指示が効果的になる可能性があります。
- 意図の理解と提案: AIがユーザーの曖昧な指示から真の意図を汲み取り、最適なアプローチや追加情報を逆に提案してくるような、より協調的なインタラクションが増えるかもしれません。
- プロンプトの自動生成・最適化: 特定のタスクに対して最適なプロンプトをAI自身が生成したり、ユーザーが作成したプロンプトを改善提案したりする技術が進化するでしょう。
- マルチモーダルプロンプティング: テキストだけでなく、画像、音声、動画などを組み合わせたプロンプトが一般的になり、よりリッチで複雑な指示が可能になります。
- エージェント化の進展: ReActのようなフレームワークはさらに洗練され、AIが自律的にタスクを計画・実行し、必要に応じて人間と対話しながら目標を達成する「AIエージェント」の活用が広がります。この場合、プロンプトはエージェントに対する高レベルな目標指示や制約条件の設定といった形になるでしょう。

しかし、どれだけAIが進化しても、「何を達成したいのか」「AIに何を期待するのか」という明確な目的意識と、それをAIに伝えるための論理的かつ効果的なコミュニケーション能力の重要性は変わりません。

プロンプトエンジニアリングは、単なる「おまじない」の羅列ではなく、AIという強力なツールとの対話術であり、その本質を理解し続けることが、未来においてもAIを最大限に活用するための

鍵となるでしょう。

おわりに

本書では、主要なプロンプト手法の概要と、2025年5月時点での代表的な生成AIモデルにおける効果について解説してきました。

生成AIの世界は日進月歩であり、今日最善とされた手法が明日には陳腐化する可能性も否定できません。しかし、各手法の基本的な考え方や、AIの特性を理解しようとする姿勢は、変化の激しいこの分野において羅針盤のような役割を果たしてくれるはずです。

重要なのは、固定観念に囚われず、実際に手を動かして様々なプロンプトを試し、AIとの対話を通じてその挙動を観察し、学び続けることです。

本書が、読者の皆様のプロンプトエンジニアリングの旅の一助となり、生成AIをより深く、より効果的に活用するための一歩を踏み出すきっかけとなれば、これに勝る喜びはありません。

以上、ご要望いただいた内容に基づき、書籍形式でまとめさせていただきました。各AIモデルの具体的なバージョン名や細かい特性は日々アップデートされるため、常に最新情報を参照されることをお勧めいたします。