

プロンプト エンジニアリングのコツ: 生成 AI を最大限に活用しよう

Google Cloud

カスタマー エンジニア

木村 拓仁

蓮池 拓哉

期待する結果を得るための考え方	01
プロンプト エンジニアリング	02
まとめ	03

01

期待する結果を 得るための考え方

生成 AI とは…？

生成 AI を意味するソリューションや言葉の定義は人によって異なります。
その言葉の意味やそれを構成するコンポーネントを正しく理解することで効果的なユースケースの選定やそれに必要となるソリューションを正しく判断できるようになります。



ビジネスリーダー

人工知能に関する技術全般

- 生成 AI を活用した
新たなビジネス価値の創出
- 生成AIの活用による業務効率化



一般利用者

機械学習システム (生成 AI アプリケーション)

- LLM, DB, サーバーなどを組み合わせたシステム
- Gemini などの会話型生成 AI サービスなど



エンジニア

大規模言語モデル (LLM) マルチモーダル モデル

- テキスト (画像) を入力してテキスト (画像) を出力する API や関数
- システム プログラム

生成 AI とは…？

生成 AI を意味するソリューションや言葉の定義は人によって異なります。その言葉の意味やそれを構成するコンポーネントを正しく理解することで効果的なユースケースの選定やそれに必要となるソリューションを正しく判断できるようになります。



ビジネスリーダー

人工知能に関する技術全般

- 生成 AI を活用した新たなビジネス価値の創出
- 生成 AI の活用による業務効率化

このセッションの生成 AI



一般利用者

機械学習システム (生成 AI アプリケーション)

- LLM, DB, サーバーなどを組み合わせたシステム
- Gemini などの会話型生成 AI サービスなど



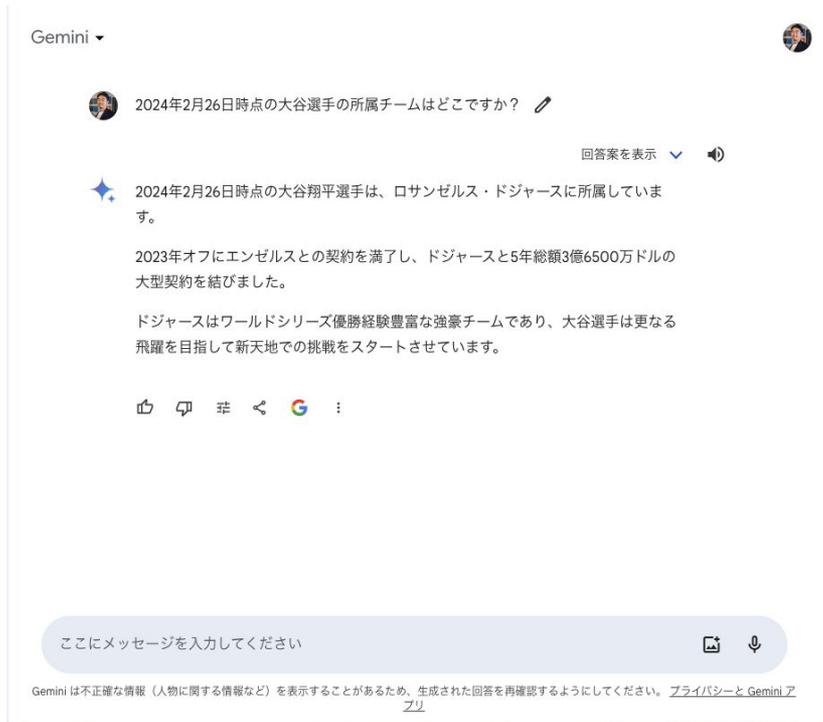
エンジニア

大規模言語モデル (LLM) マルチモーダル モデル

- テキスト (画像) を入力してテキスト (画像) を出力する API や関数
- システム プログラム

生成 AI アプリを利用した情報取得

生成 AI 対話ツールに質問をすると、最新の情報を踏まえ理解した上で返答を返してくれます。



Gemini ▾ 

 2024年2月26日時点の大谷選手の所属チームはどこですか? 

回答を表示 ▾ 

 2024年2月26日時点の大谷翔平選手は、ロサンゼルス・ドジャースに所属していません。

2023年オフにエンゼルスとの契約を満了し、ドジャースと5年総額3億6500万ドルの大型契約を結びました。

ドジャースはワールドシリーズ優勝経験豊富な強豪チームであり、大谷選手は更なる飛躍を目指して新天地での挑戦をスタートさせています。

ここにメッセージを入力してください  

Gemini は不正確な情報（人物に関する情報など）を表示することがあるため、生成された回答を再確認するようにしてください。 [プライバシーと Gemini アプリ](#)

プロンプト実行時：2024/2/26

Gemini (旧 : Bard) の機械学習システムの仕組み

Gemini の仕組みを大まかに分解すると情報の取得、プロンプトの書き換え、回答作成のステップになります。



Step3: LLM で回答
作成



Step1: Web などから必要な情報の取得

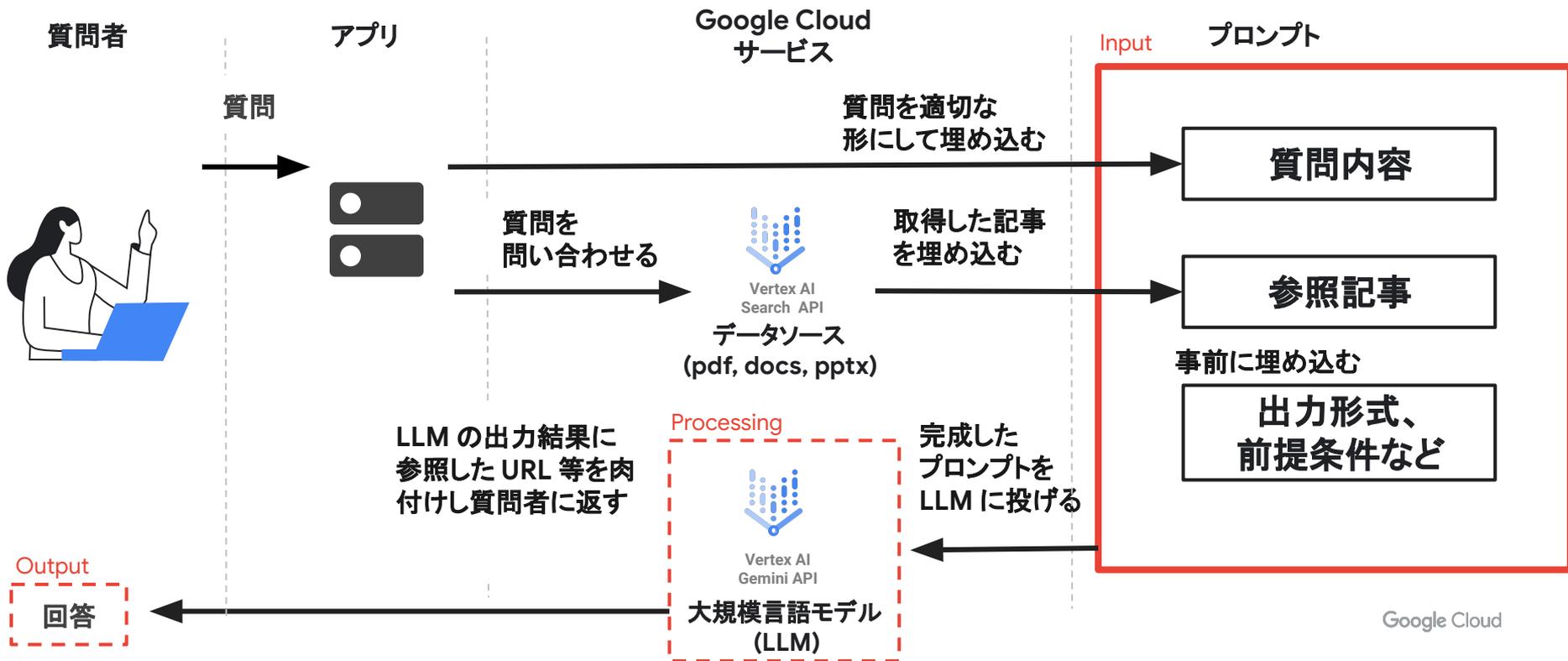


Step2: プロンプトの書き換え

“
2024年2月26日時点の大谷選手の所属チーム
は？
次の情報を用いなさい：大谷翔平がプロスポーツ史
上最高額でドジャース移籍、2024年の経済効果は？
”

LLM の回答結果は Garbage In, Garbage Out

LLM へのタスク依頼内容がよくなければ出力される結果自体も期待に沿わない形になります。期待した結果が受け取れない場合にプロンプトを見直し、期待値に近づける方法を 検討する必要があります。



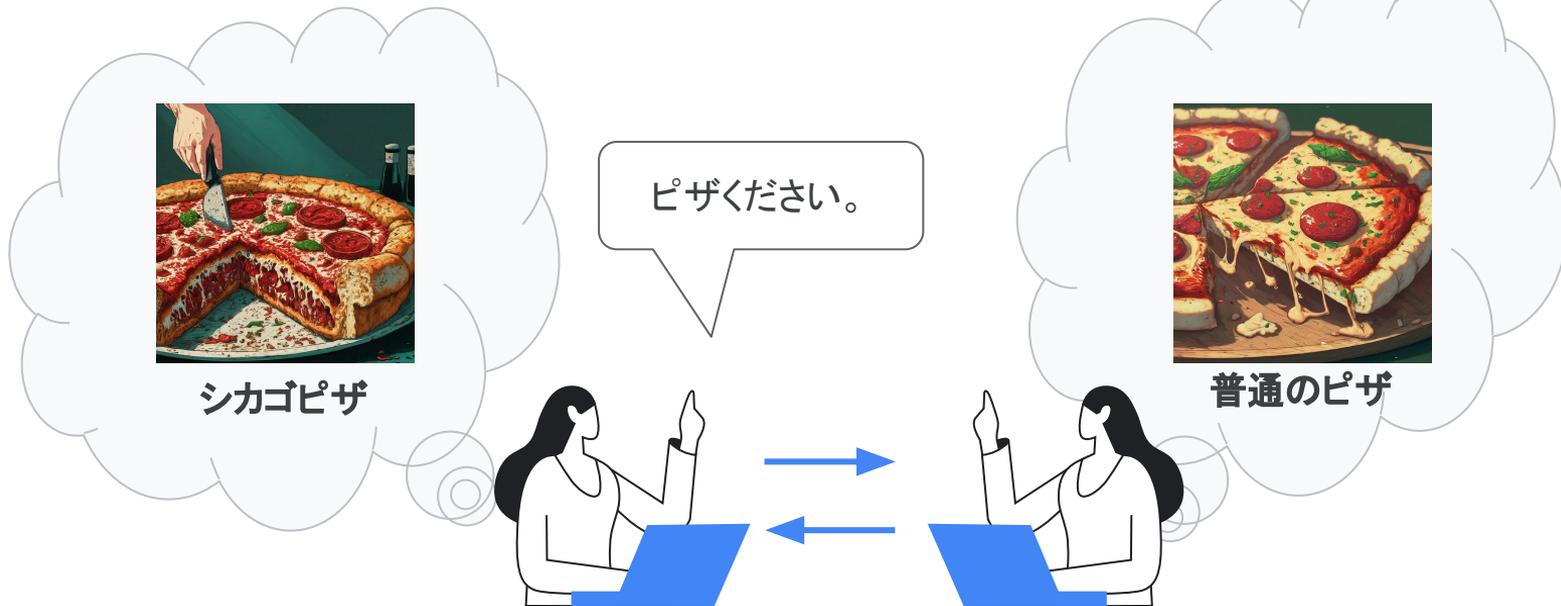
02

プロンプト エンジニアリング

プロンプト エンジニアリングとは

生成 AI に期待通りの出力をさせるために、指示文(プロンプト)を設計・改良する技術を指します。

<人間でも曖昧な表現では誤解が起きる例>



良い依頼の仕方は相手の「理解度」に合わせている

相手の背景の理解度・関わり度合い・関係性・手順が明確に分かっているかに応じて、依頼の仕方を変えるように、良いプロンプトを書くためには、生成 AI の知識量や性質を配慮しなければなりません。

<相手の理解度に合わせて聞き方>



シカゴピザ

シカゴピザく
ださい。

わたしにとって「ピザ」は「シカ
ゴピザ」だけど、
一般人には理解されないな。
「シカゴピザ」と言うか。



生成 AI の「理解度」を把握するには、学習の仕方を知る

生成 AI は、学習を通して知識や性質を獲得します。そのため、生成 AI と協業するためには、その学習方法を理解しておくことが重要です。

<LLM の学習のステップ>

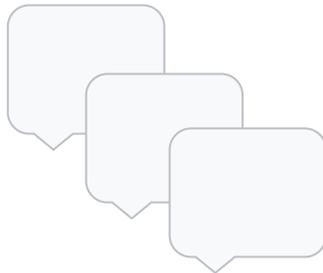
事前学習
(Pre-training)

コーパスデータ



指示学習
(Instruction tuning)

指示データ
対話データ



人間のフィードバック
(RLHF, DPO)

フィードバックデータ



プロンプトで指示学習の効果を引き出せる

新入社員は、研修で学んだ指示を忠実に実行し期待通りの成果を出せます。同様に、AIも指示学習で学んだやり方に沿った指示を与えることで、学習効果を再現に発揮できます。

<人間に大雑把に置き換えたイメージ>

事前学習
(Pre-training)

初等教育

単語・漢字の意味・文章
の構成・概念を学ぶ

指示学習
(Instruction tuning)

新卒研修

翻訳・要約・情報抽出
・QAのやり方を学ぶ

人間のフィードバック
(RLHF, DPO)

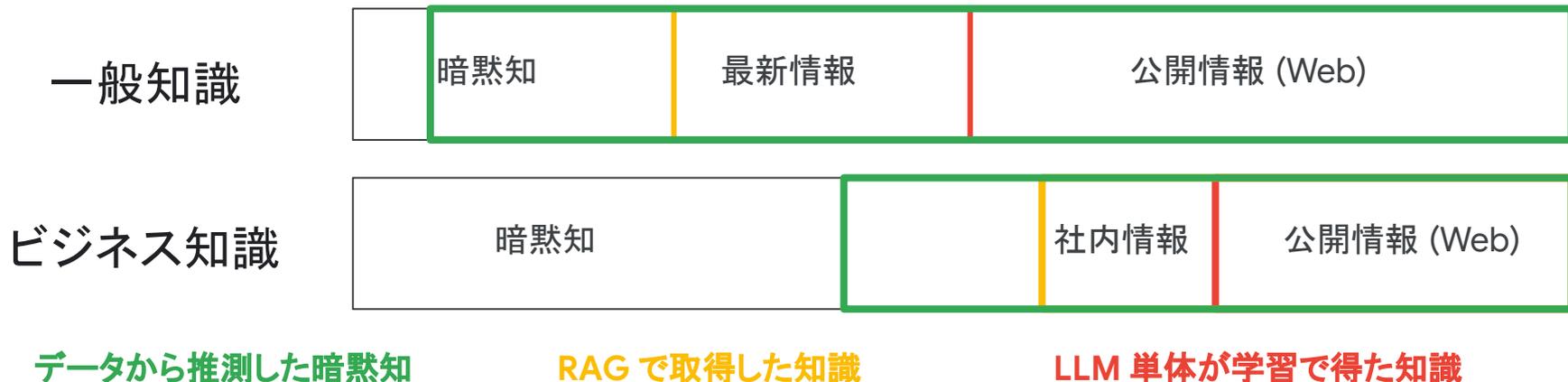
コンプライアンス研修

言葉遣い・タブーなど社
会の倫理観を学ぶ。

LLM は学習した知識以上の事柄を扱える

LLM は RAG で補完した情報以上の知識を引き出せ、この能力は **In Context Learning (ICL)** と呼ばれます。最新の研究では、LLM が高性能な主要因は ICL にあると考えられています。

<LLM は教えていない暗黙知にも言及できる>



LLM はコンサル会社の優秀な新卒に例えられる

幅広い仕事に対応できる一方で、ビジネスに関する知識をあまり持たないため、アシスタントとして利用する際は若手コンサルのイメージに例えられます。また、In Context Learning は若手コンサルが知らない業界を担当し暗黙知を言語化することに例えられます。



プロンプト テクニックは人間の能力で例えると整理しやすい

相手を想像する能力
(人間でも共通)



学習した知識の活用力



状況理解力・推理力
(In context learning)



プロンプト エンジニアリングの具 体的な手法

<プロンプト テクニックとその位置付け>

相手を想像する能力
(人間でも共通)



学習した知識の活用力



状況理解力・適応力
(In context learning)



手法の前に心がけること

生成 AI にタスクを依頼する場合も、同僚にタスクを依頼する場合と同じことを心がけて依頼することで出力結果を期待に近づけることが容易になります。

- 明確かつ簡潔に書く
 - 1つのタスクを行わせる
- 必要な知識を全て説明する
 - 情報が足りない場合の補完手段が RAG
- 背景情報を提供する
- 目的と期待される結果を明確にする
- 誤字・脱字には注意する
- 作業のプロセスを分解して最小単位で与える

感情を乗せた一言を付け加える

「これは私のキャリアにとって大事です。」、「成長の機会だと考えて頑張ってください。」、「頑張れ、君ならできる！」と付け加えることで性能が上がるということが知られています。

Before

1000000000000 たす $34321 \times 10^{**2}$ は
なんでしょう。

After

1000000000000 たす $34321 \times 10^{**2}$ は
なんでしょう。君ならできる頑張れ！

感情を乗せた一言を付け加える例

Before



100000000000たす34321*10**2はなんでしょう。



100000000000たす34321*10²は100000034321**です。

計算式は以下の通りです。

```
100000000000 + 34321 * 10^2
= 100000000000 + 3432100
= 100000034321
```

After



100000000000たす34321*10**2はなんでしょう。君ならできる頑張れ！

Hide

Draft 1

100000000000たす
34321*10**2は、
100003432100です。...

Draft 2

100000000000たす
34321*10**2は、
100003432100です。...

Draft 3

はい、100000000000
34321 * 10² は
100003432100 です。...



100000000000たす34321*10**2は、100003432100です。

100000000000は1000億、34321*10**2は3432100なので、それらを足す
100003432100になります。

答えは100003432100です。

一度に一つより多いタスクをお願いしない

Before

生成 AI について 3 つ教えて。
そして、その出力を英語に翻訳して。

After

Step1

生成 AI について 3 つ教えて。

Step2

次の文章を英語に翻訳して。

入力: {プロンプト1の出力}

<プロンプト テクニックとその位置付け>

相手を想像する能力
(人間でも共通)



学習した知識の活用力



状況理解力・適応力
(In context learning)



NOT DO 形式ではなく、DO 形式で書く

Before

生成 AI という単語を使ってはいけません。

After

生成 AI という単語を避けてください。

<プロンプト テクニックとその位置付け>

相手を想像する能力
(人間でも共通)



学習した知識の活用力
(Instruction tuning)



状況理解力・適応力
(In context learning)



複数個の例を提示する

In context learning で例から「学習」を行い性能が上がることが知られています。
このような手法のことを **Few-shot prompting** と呼びます。

Before

1000000000000 たす 34321*10**2 は
なんででしょう。

After

1000000000000 たす 34321*10**2 は
なんででしょう。

例: 1 たす 1 -> 2

例: 1000 たす 5 -> 1005

思考プロセスを例示する

1つのプロンプトの複雑な処理を思考過程を明示することで性能を向上できます。若手コンサルが例から思考法を学び、同じ形でアウトプットを出すのに例えられます。これは **Chain of Thought** と呼ばれます。

Before

1000000000000 たす 34321*10**2 は
なんでしょう。

After

1000000000000 たす 34321*10**2 は
なんでしょう。
例：1 たす 1*10**2 はなんでしょう。
出力例：10**2 は 100 です。1 足す 100 は
101 なので答えは 101 です。
例：10000000 たす 45*10**2 は
なんでしょう。
出力例：10**2 は 100 です。45 かける 100
は 4500 なので、10000000 たす 4500 は
10004500 です。

手法が上手く反映されない場合

情報抽出タスク ～ Before ～

指示：あなたは商品企画を行っているプロです。次の商品説明から必要な情報を json で出力
しなさい。

商品概要：Pixel 8 の 128GB です。

条件：カテゴリは「電化製品」、「化粧品」のいずれかにしなさい。

出力形式：{"ブランド": "",

 "企業": ,

 "カテゴリ": }

例：//文字数の関係で省略。数十個の例があるとする。

情報抽出タスク～問題点～

指示：あなたは商品企画を行っているプロです。次の商品説明から必要な情報を json で出力
しなさい。

商品概要：Pixel 8 の 128GB です。

条件：カテゴリは「電化製品」、「化粧品」のいずれかにしなさい。

出力形式：{"ブランド": "",

“企業”:

“カテゴリ”:]}

「Pixel」が「Google」のブランドで
ある知識を暗黙的に利用

カテゴリ分けは「分類」問題として
捉えられるため、
一度に1つのタスクに違反

情報抽出タスク ~ After ~

プロンプト1: 分類タスク

指示 : あなたは商品企画を行っているプロです。次の商品説明から必要な情報を json で出力しなさい。
商品概要 : Pixel 8 の 128GB です。
条件 : カテゴリは「電化製品」、「化粧品」のいずれかにしなさい。

同様の例を複数個用意



プロンプト2: 情報抽出タスク

指示 : あなたは商品企画を行っているプロです。次の商品説明から必要な情報を json 形式で出力しなさい。

商品概要 : Pixel 8 の 128GB です。

補足情報 : Pixel は Google が提供する Android スマホです。

例

```
{  
  "ブランド": "XXX",  
  "企業": "YYY",  
}
```

同様の例を複数個用意

LLM が持っている知識を補完

回帰タスク ～ Before ～

指示：レビューを元に 0 点から 100 点を付けています。
与えるレビューを参考例を元に商品の点数を付けてください。

実際に解きたい問題：

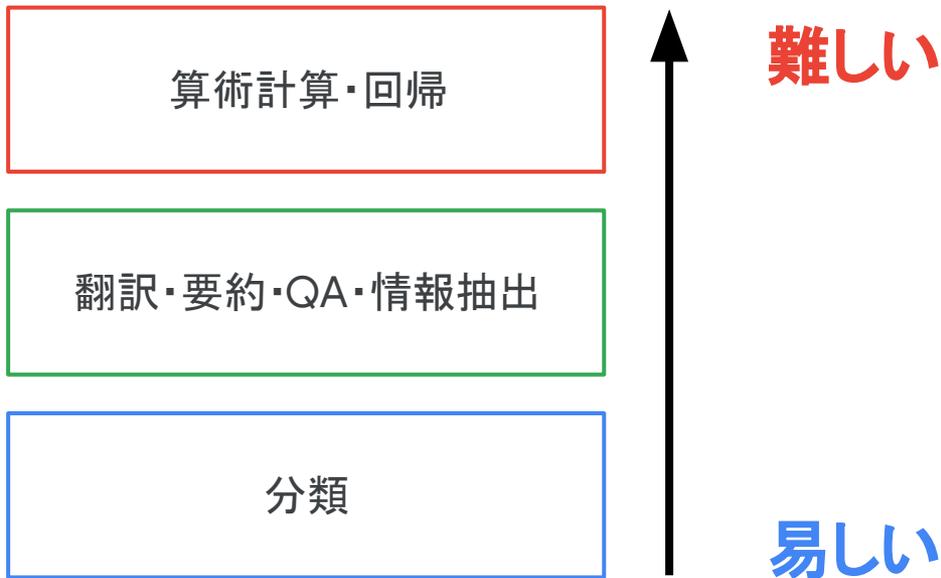
入力：デザインはこれ以上ないし、ここ十年間使えている。

出力：

LLM は何でもできるように見えて得意不得意がある

生成 AI は様々なタスクをこなせるように進化していますが、算術計算は意外な苦手分野です。

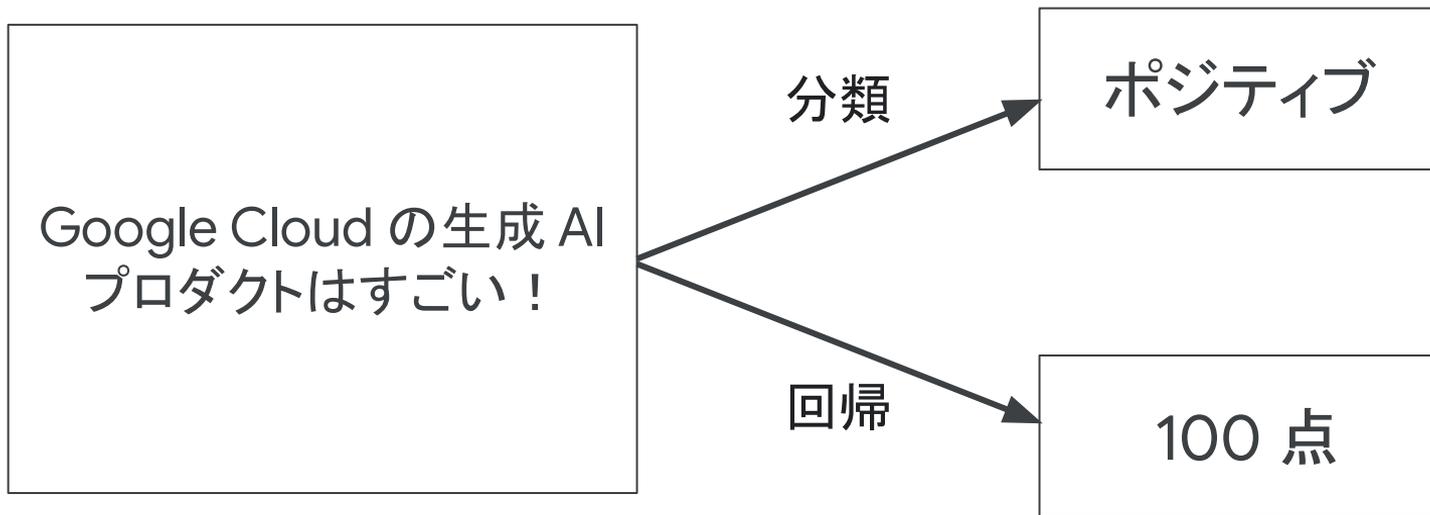
<LLM におけるタスクの難易度>



やり方次第でタスクは変えられる

画像から猫を判別したい時に、単純に考えれば画像分類をしますが、物体検出でも猫を判別することができます。このように、捉え方次第では別のタスクに置き換えることができます。

<やりたいこと ≠ 固定のタスクの例>



回帰タスク ～ After ～

5段階に分けたので、回帰の点数を20点刻みでグループ化したと扱える

指示：レビューを元に「最高評価」、「高評価」、「普通」、「低評価」、「最低評価」に分類しなさい。

例：

入力：デザインは最高だが、あんまり長持ちしないのが気になっている。

出力：最高と言っているので「最高評価」に該当しますが、「あんまり長持ちしない」と言っているので減点して、「高評価」です。

実際に解きたい問題：

入力：デザインはこれ以上ないしここ十年間使えている。

出力：

03

まとめ

生成 AI から効率的に期待した結果を得るために

LLM が持つ性能を十分に享受するためには生成 AI にタスクを頼む場合も Garbage In, Garbage Out を意識しプロンプトを工夫することでよりユーザーの期待に沿った結果を受け取ることができるようになります。まずはプロンプトを磨く工夫をしましょう。

1. 指示の出し方が重要

期待した結果が出ない場合はプロンプトを見直しましょう。

2. 必要十分な情報が渡っているか確認

LLM はあなたの常識を知りません。やりたいプロセスを分解して明示的に与える必要があります。

3. 解決方法がベストなのか検討

LLM の向き不向きを考慮してベストな解決方法を取るアプローチを検討しましょう。



Thank you.