

aa-14. 対話型 AI

(コンピューターサイエンス)

金子邦彦

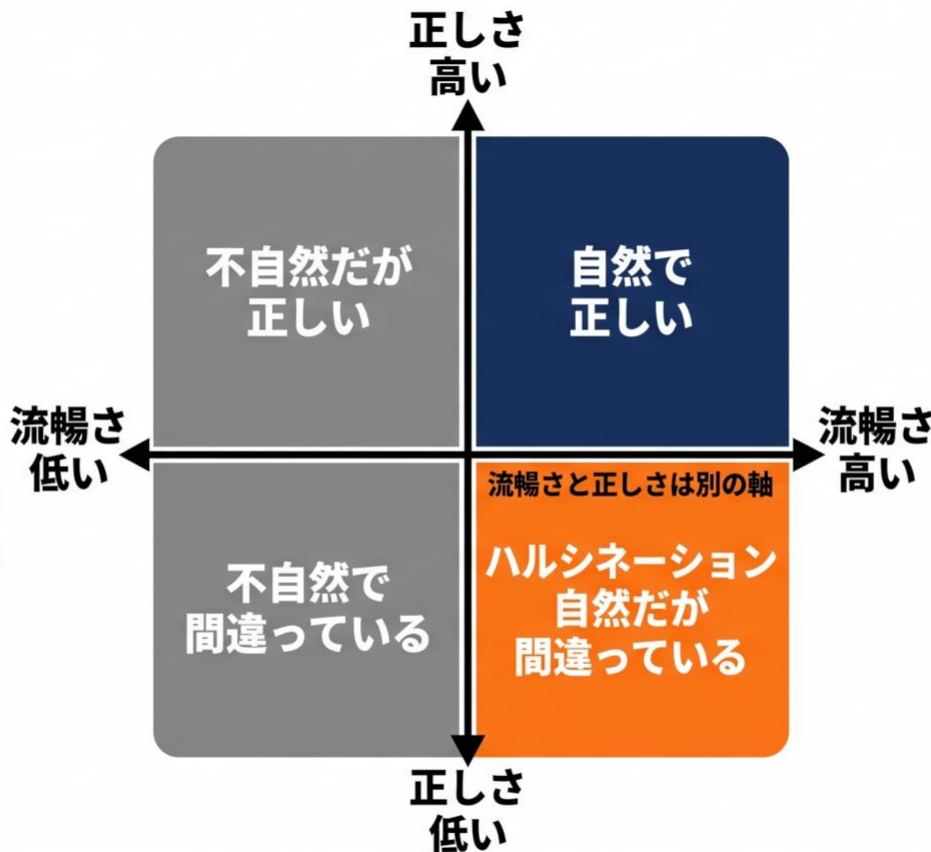
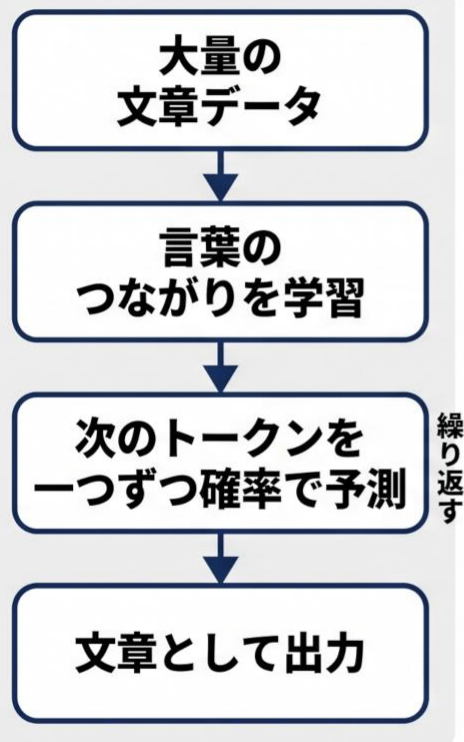


「人工知能」第14回の内容



対話型AIの文章生成は確率予測であり、流暢さと正しさは別問題である

AIはどう文章を作るか



プロンプトの工夫

Web検索・
外部情報の参照
(グラウンディング)

コード生成・実行

AIの回答が
改善する場合がある

正しさを保証する
ものではない

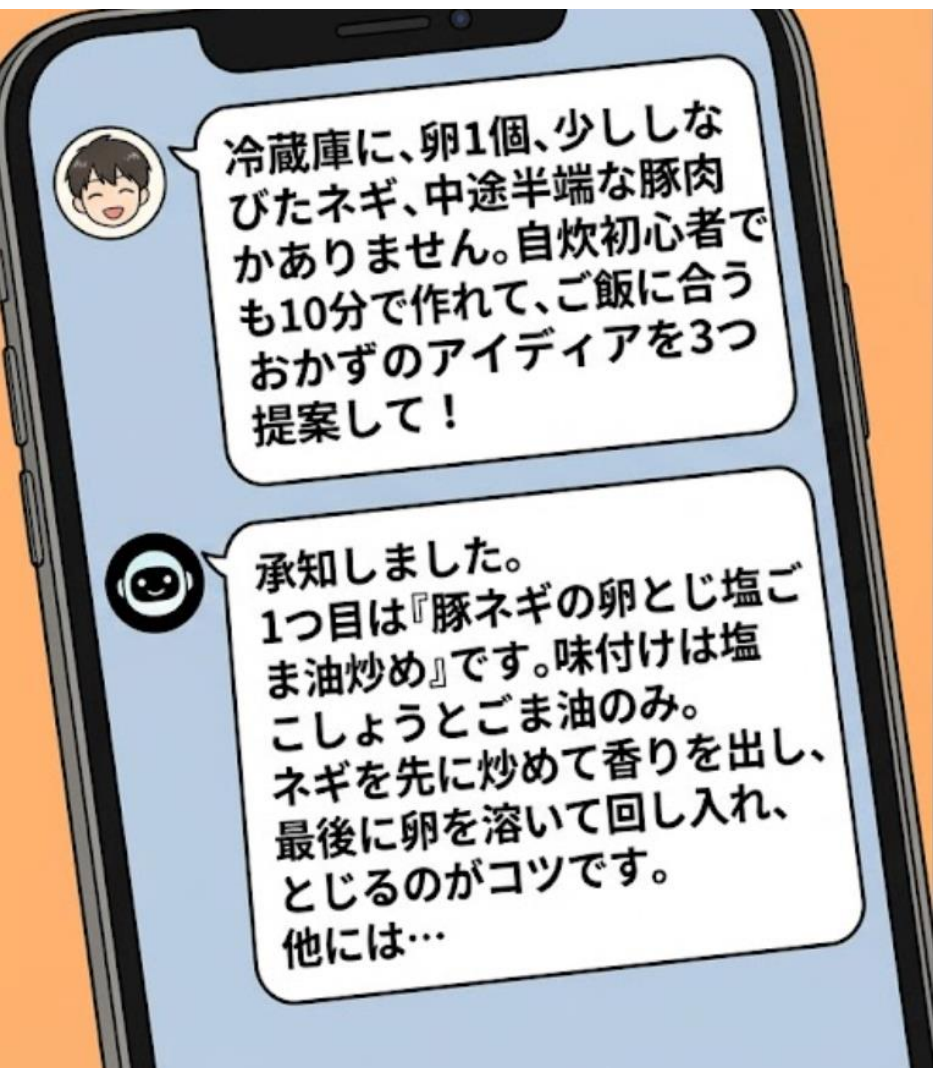
たたき台の作成や
作業の自動化に活用する

個人情報・機密情報の
入力は避ける

最終確認は必ず
人間が行う



14-1. 対話型AIの仕組み





Copilotのようなチャット型AIは、質問に対して自然な文章で応答を返す。この応答は、内部で『大規模言語モデル（LLM）』と呼ばれる仕組みによって生成されている。

この“自然な応答”は、どのような仕組みで作られているのか？
一次のページで、その仕組みの中心にある『LLM』と『トークン』を見ていく



AI の回答. こんな経験はありませんか？



聞き方を変えたら、答えの質が変わった



自信を持って答えてくれたが、
本当に正しいのだろうか

なぜ、このようなことが起こるのだろうか？

この資料では、AIの“仕組み”を理解することで、こうした“振る舞い”の理由を読み解いていく

大規模言語モデルとトークンとは



大規模言語モデル (LLM: Large Language Model)

直前の文脈から、次に来やすいトークンを確率的に予測し、それを繰り返して文章を生成するモデル。
テキストデータで学習した言語モデルである。



文脈→予測→追加→繰り返し

トークンとは

LLMが予測・処理する単位であり、単語そのものではなく、単語または単語の一部に相当する。

対話型AI → 対話 型 AI

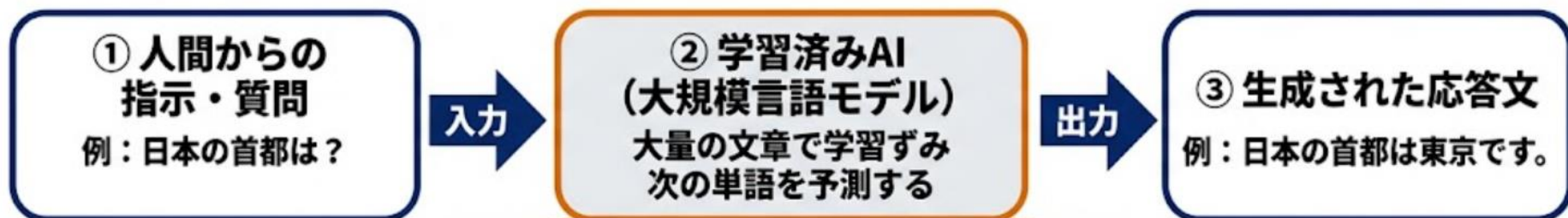
※トークンの分割方法はモデルにより異なる

※英語では tokenization → token / ization のように"単語の一部"に分かれることもある

対話型 AI の対話の仕組み



対話型AIは、入力された文をもとに『次に来る単語』を確率で予測し、1語ずつつなげて応答文を作る



② の中で起きていること

1. 文を小さな単位に分ける

日本 の 首都 は ？

2. 次の単語の候補を確率で計算



3. いちばん確からしい語を選んで文末に足す

日本の首都は → 東京

文が完成するまで、この予測を1語ずつくり返す

AIは答えを検索しているのではなく、学習した膨大な文章の統計から『続きの言葉』を予測している

プロンプトと出力の関係 — なぜ聞き方で答えが変わるのか

AIの予測は直前の文脈全体を手がかりに行われる — 人間が入力するプロンプトは、その文脈の一部である

入力例：『AIについて教えて』



文脈に含まれる情報が少ない（対象や範囲が曖昧）



次に来やすいトークンの候補が幅広く分散する



出力：一般的・曖昧な内容になりやすい

入力例：『生成AIの仕組みを3つの観点で説明して』



文脈に含まれる情報が多い（対象や範囲が明確）



次に来やすいトークンの候補が、
関連性の高いものに絞られる

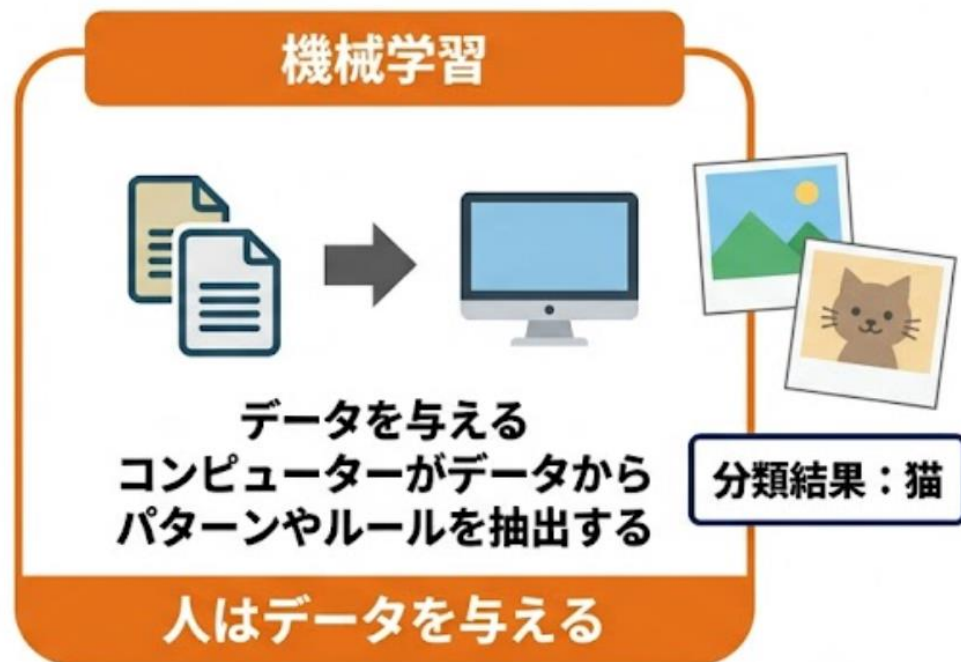


出力：意図に沿った具体的な内容になりやすい

注意：プロンプトに「正確に回答せよ」と書いても、正確性が保証されるわけではない。プロンプトを具体的・詳細にすることで回答は変化するが、「正しさ」は別のことである



機械学習とは、ルールを人が教えるのではなく、データからコンピューター自身にパターンやルールを見つけさせる仕組みである



機械学習で人が与えるのはデータであり、ルールそのものではない

対話型AIの仕組み — 大量のデータから"言葉のつながり"を学ぶ

① 大量の文章（本・Web記事・会話など）を教材として集める



② AIが「どのトークンの次に、どのトークンが来やすいか」の規則性を学ぶ



③ 学んだ傾向は膨大な"数値の設定"としてAIの中に蓄積される



④ 質問には、その傾向をもとに"次に来やすいトークン"を一つずつ選んで文章を作る

注意：AIは人間のように"意味"を理解しているわけではなく、言葉のつながりの確率をもとに文章を作っています

対話型AIは、学んだ「ことばのつながりの傾向」をもとに、次のことばを1つずつ予測して文をつくる

AIが持っているもの

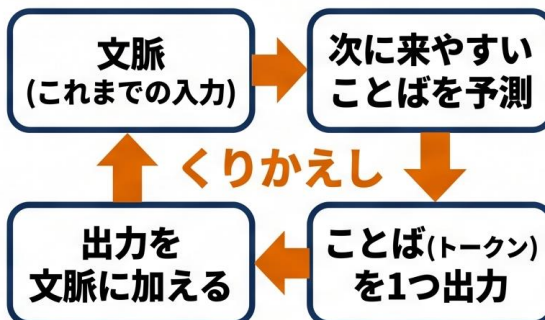
「大量の文章データ」

↓ 学習

「ことばのつながりの傾向 (確率)」

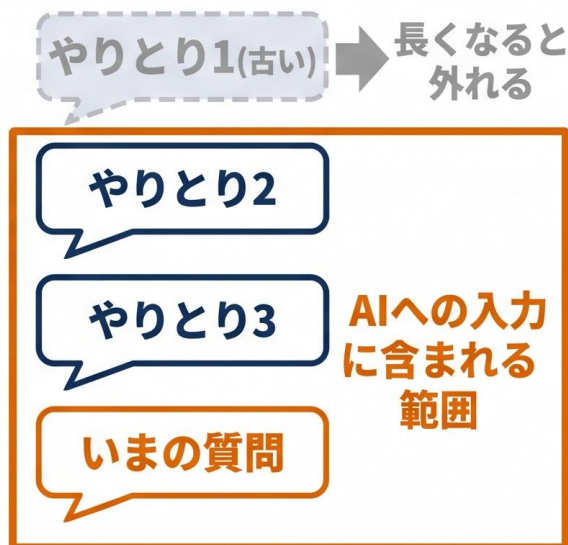
今日は	天気	55%
いい	日	20%
話		8%

1つずつ予測してつなげる



ことばを → 1つ
ことばを 1つ → ずっと
ことばを 1つずつ → 予測

文脈はチャット全体



AIは人間のようにことばの意味を理解しているわけではない。

対話型AIの特質：「流暢さ」と「正しさ」は別の軸で動く



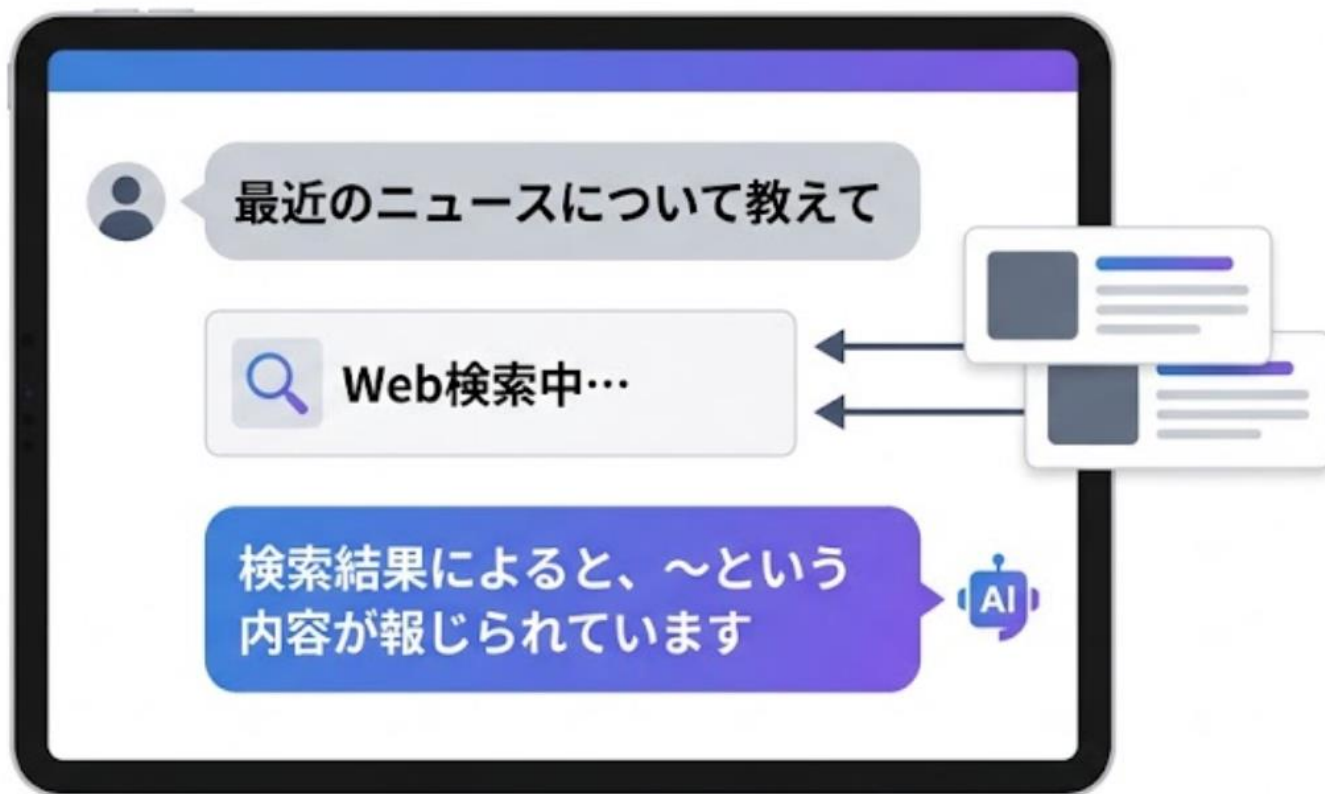
文章の自然さ（流暢さ）と内容の正しさ（正確さ）



ハルシネーションとは、AIが誤った内容を、自信を持って正しいかのように述べてしまう現象を指す



Web検索による情報補完



AIは自身が学習した情報だけでなく、必要に応じてWeb検索を行い、その結果を回答に反映することがある

イメージ：AIによるプログラム生成と実行

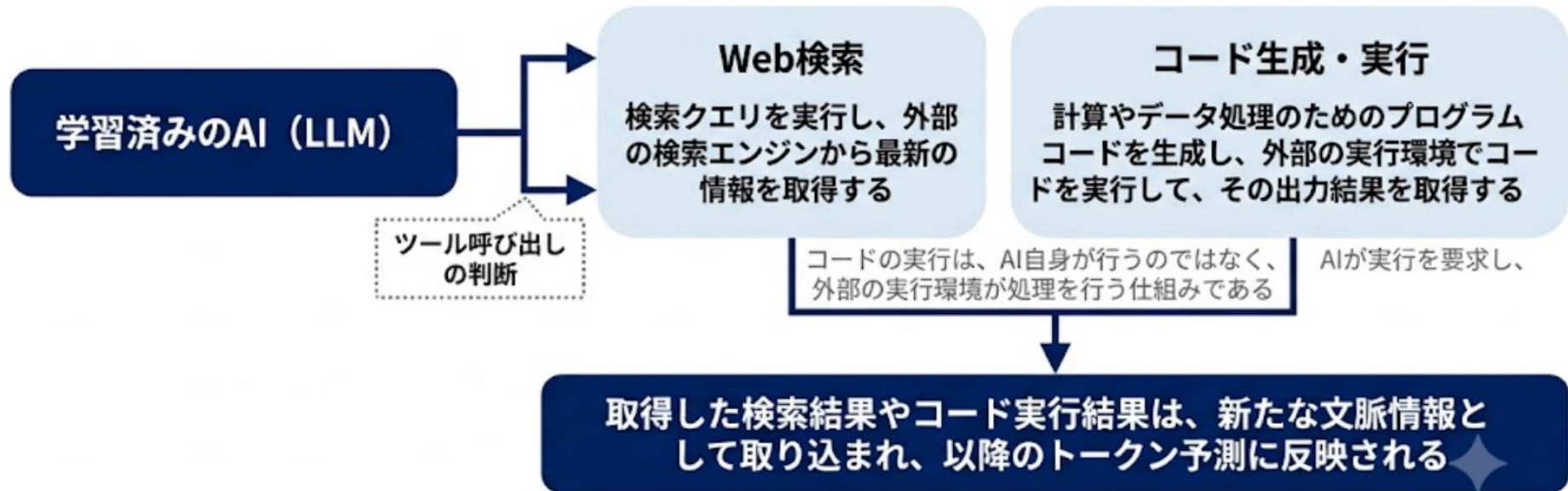


AIはプログラムコードを生成し、外部の実行環境でそのコードを動かして結果を得ることがある。コードの実行はAI自身の内部ではなく、外部の環境で行われる

AI による外部ツール（Web検索、プログラム生成と実行）の活用



対話型AIは、必要に応じて外部のツールを呼び出し、その結果を文章生成に利用することがある



注意：ツールを利用しても、最終的な文章はトークンの確率的予測によって生成されるため、内容の正しさが保証されるわけではない



14-2. 対話型AIの活用



ここ行ってみよう！
AIのおすすめだって

え…
そこもう潰れて、
今は別のお店に
なってるよ？

駅前でおすすめの
『トラットリア○○』
です！

AI の回答は次の3点で注意が必要



指示（プロンプト）の出し方で答えは変わり、時には自信满满だが事実や実際の内容と異なる回答が起こることもある。

プロンプトの工夫で 結果が変わる

『AIについて教えて』
(ざっくりとした指示)



曖昧で漠然とした回答

『生成AIのしくみを
3つのポイントで説明して』
(具体的な指示)



整理された明確な回答

指示の出し方で
答えの質が大きく変わる

AIは自信满满に間違えること もある (ハルシネーション)

※以下はハルシネーションを理解するための、
わかりやすくした簡略化例です

富士山の高さは？

5000メートルです

実際の正しい高さ：3776メートル



もっともらしく自信满满に答えるが、
事実とは異なることがある
＝ハルシネーション

※AIは事実を検索しているのではなく、学習データのパターン
から次に来る言葉（トークン）を統計的に予測して文章を作
っている。そのため誤りにも気づかず自信满满に答えてしまう

実際の内容とズレた提案を してしまうこともある

『Python入門の授業
レポートを考えて』



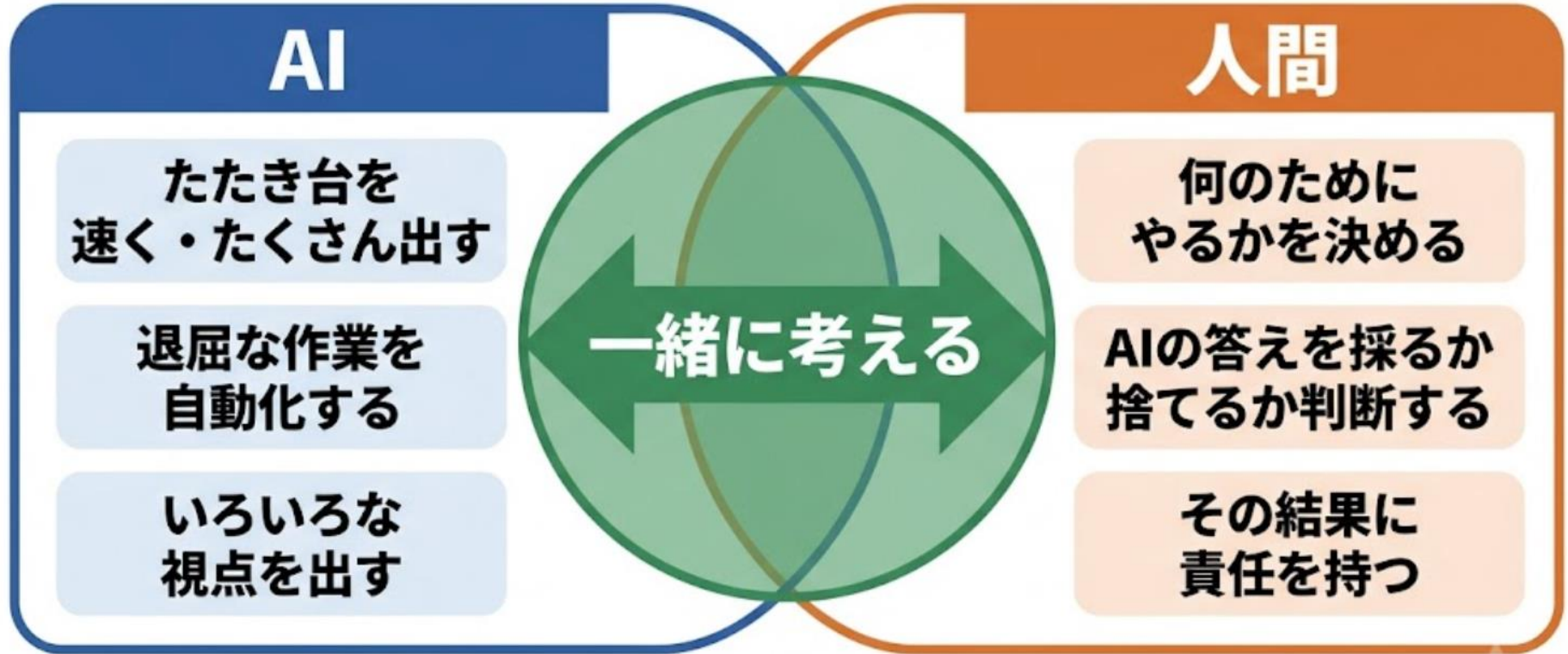
『tryは信頼性向上に役立つ』
など、授業でまだ習っていない
内容を含めてしまう

実際の授業で習った内容について、
自分で考えて作成することで
実力が伸びるもの



AIはあくまで相談相手であり、
正解をそのまま出す存在ではない。
内容は自分で確認・調整する

AIの得意と人間の役割を分けて、一緒に考える



AIに考えさせるのではなく、AIと一緒に考える



AIを使うときは、情報の『入れ方』と『答えの使い方』に注意が必要



AIに入れてはいけない情報

- 自分や他人の個人情報
- 他人の秘密
- 公開されていない機密情報

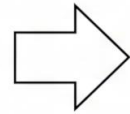
AIの答えを使うときに守ること

- AIの回答をそのまま信じない
- 最終的判断は人間の役割

AIの結論を『たたき台』から『検証済みの結論』にする

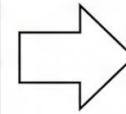
AIの結論（たたき台）

例：富士山の高さは5000メートルです



検証の3つの視点

- 1 根拠は一次資料までたどれるか
- 2 自分が知っている事実や他の資料と一致するか
(例：富士山の正しい高さは3776メートル。食い違うためハルシネーションの可能性がある)
- 3 その情報はいつの時点のものか



検証済みの結論

根拠を確認し、採用すると人間が判断した結論

根拠を確認するまで、AIの結論は『たたき台』のまま



自分で試す × AIに相談する × チームで対話する — 研究も仕事も、この3つで前に進む

この進め方は、将来の仕事でこそ役立つ

AIアシスタントとプログラミング

AIアシスタントの例 — Colabは登録無料

やりたいことを言葉で伝えれば、AIがコードを書くのを手伝ってくれる

コードを書く場所

```
▶ import plotly.graph_objects as go
import numpy as np

np.random.seed(42)
n = 100
x = np.random.randn(n)
y = np.random.randn(n)
z = np.random.randn(n)

fig = go.Figure(data=[go.Scatter3d(x=x,
                                   y=y, z=z, mode='markers')])
fig.show()
```

↔
相談しながら
コードを完成

AIアシスタント (Gemini)

こんにちは。ご用件をお聞かせください

Pythonライブラリのインストール方法

Googleドライブからデータを読み込む

シンプルなMLモデルの例

プログラミングが苦手でも安心。コードは自力で解くものではなく、自分の道具。
『動いた』を一つずつ積み重ねよう

Google Colab の AI アシスタント

コード編集・実行画面

AIアシスタント画面

Google Colab の AI アシスタント画面のスクリーンショット。左側はコード編集・実行画面、右側は AI アシスタント画面（Gemini）が表示されている。

コード編集・実行画面（左側）:

- タイトル: Untitled14.ipynb
- メニュー: ファイル、編集、表示、挿入、ランタイム、ツール、ヘルプ
- 検索: コマンド
- アクション: + コード、+ テキスト、すべてのセルを実行
- ステータス: 接続
- メッセージ: [] コーディングを開始するか、AI で生成します。

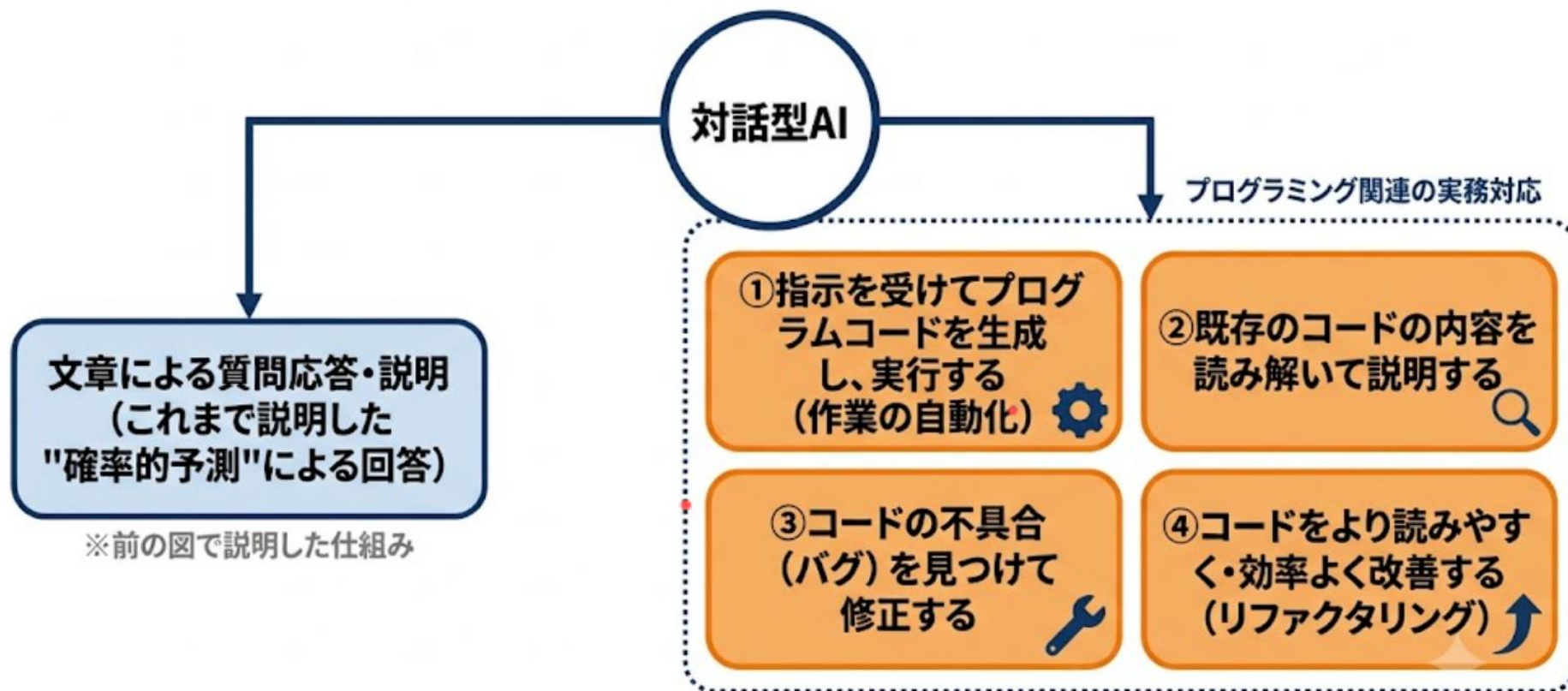
AI アシスタント画面（右側）:

- モデル: Gemini
- 挨拶: こんにちは、邦彦さん
- メッセージ: ご用件をお聞かせください。
- リンク:
 - Python ライブラリをインストールする方法
 - Google ドライブからデータを読み込む
 - シンプルな ML モデルのトレーニングの例
- 入力欄: 何を構築しますか？
- モデル選択: Gemini 2.5 Flash

プログラミングでの対話型AIの活用



対話型AIの役割は"会話による回答"だけではない——コードの生成・実行、そして相談にも対応する



結論：対話型AIは"会話の相手"であると同時に、指示に応じて動く"実務ツール"としても機能する



コードは動くが、「6～12歳の子供料金」の区分が抜けている（10歳が通常料金になる）。確認・修正が必要



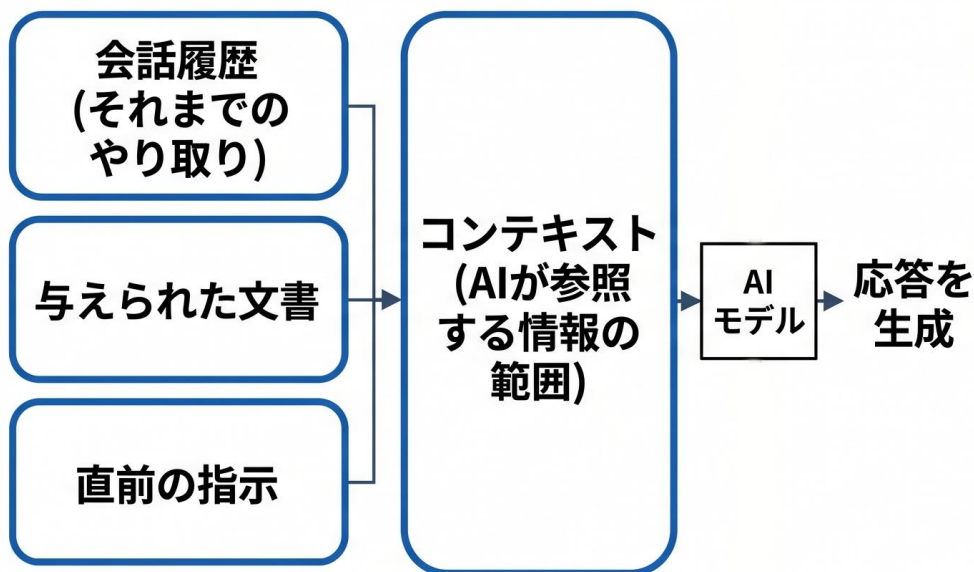
14-3. 対話型AIの様々な能力・機能

① 会話履歴や文書等にも対応（コンテキスト）

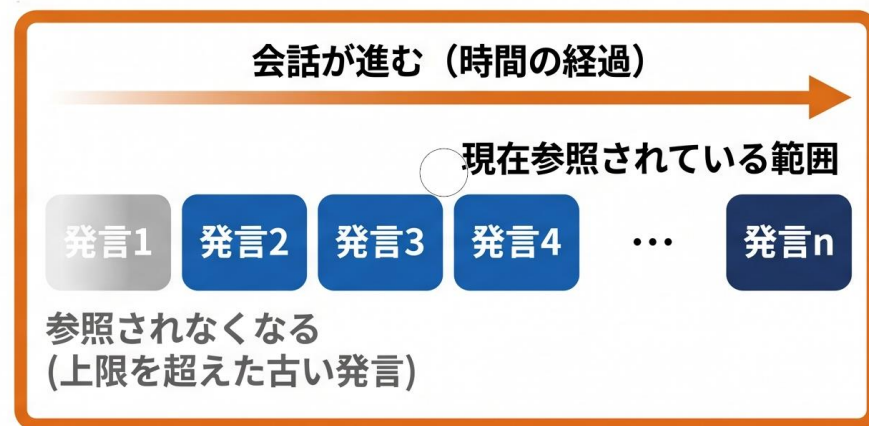


AIは直前の指示だけでなく会話履歴や文書も参照して応答するが、参照できる情報量には上限がある

コンテキストの構成



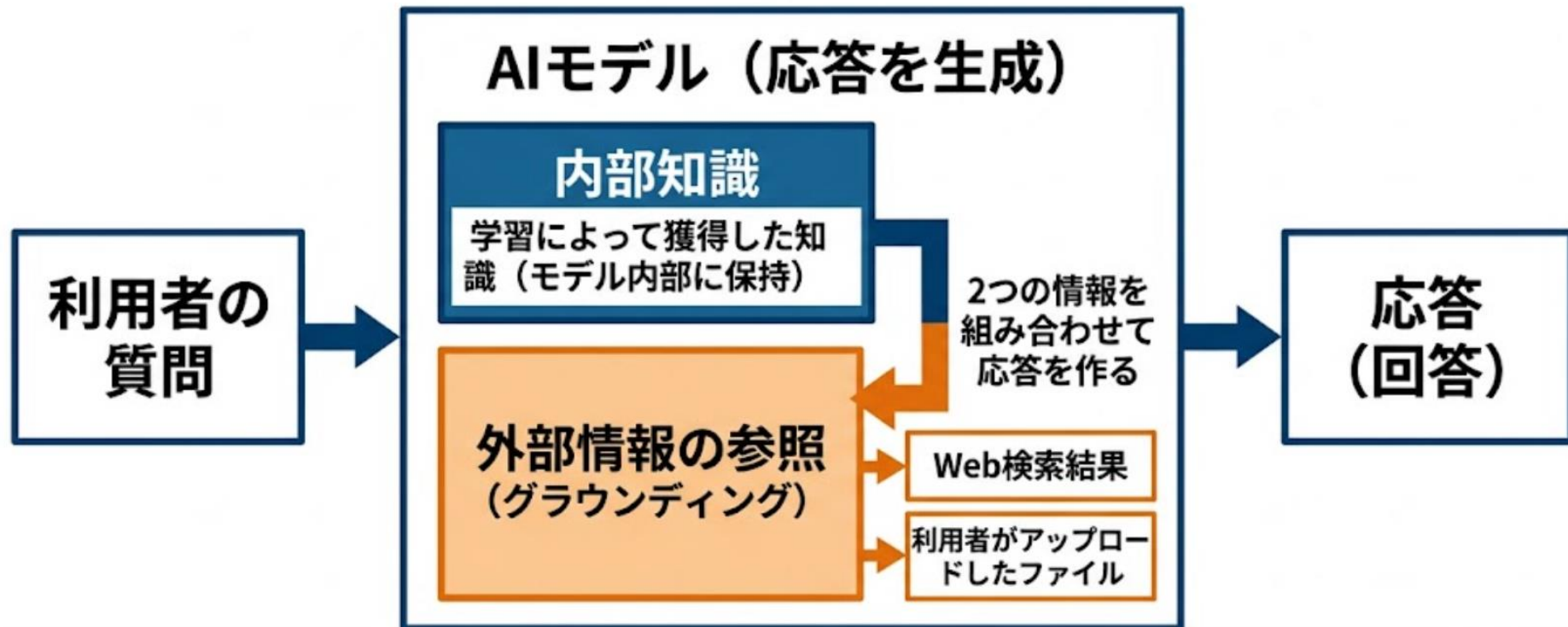
コンテキストウィンドウ（参照できる量の上限）



② 外部情報の参照（グラウンディング）



AIは内部の知識だけでなく、外部の情報も参照して応答を作る



外部情報
が誤り

応答も
誤り

注意：参照先の外部情報が誤っている場合、
応答も誤ったものになり得る

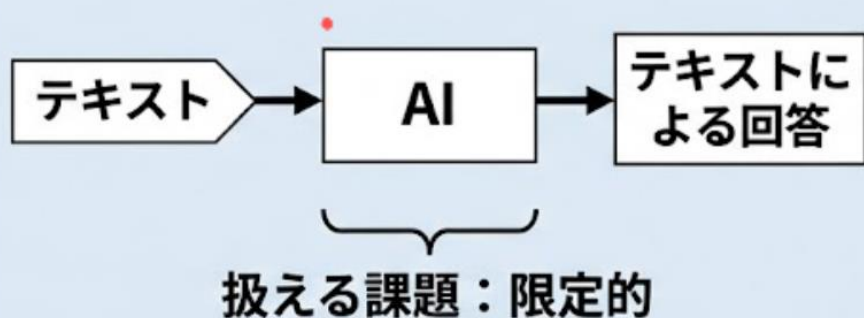
③ 画像や文書や表なども扱えるマルチモーダルAI



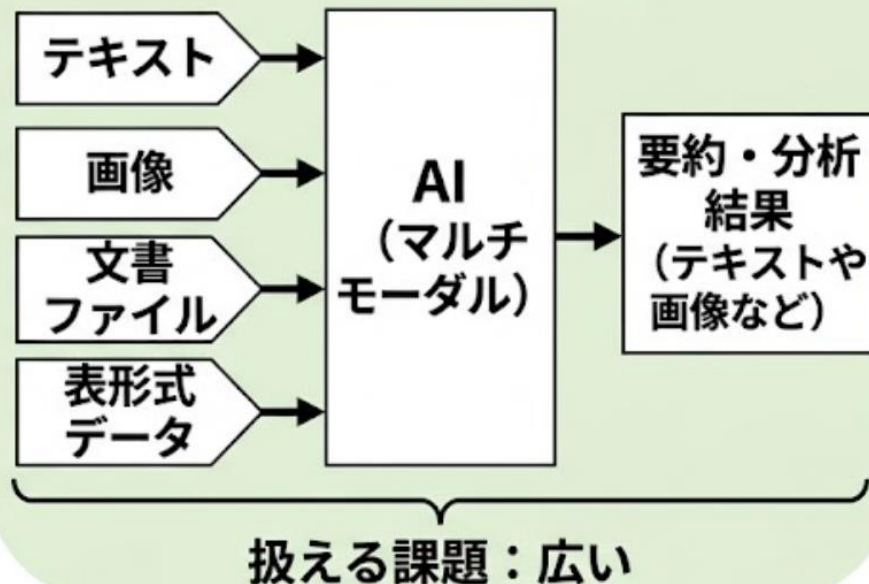
モーダル＝入出力の種類。

複数種類を扱えるAIをマルチモーダルAIと呼ぶ

テキストのみのAI



マルチモーダルAI



例：Microsoft 365 Copilot



④ データファイルの中身を読み取り処理する機能



Excelの関数を覚えなくても、自然な言葉で指示するだけでデータの集計・分析ができる

従来の方法

Excelの表データ



SUM関数・VLOOKUP関数などを自分で入力する



集計やグラフ作成に時間と知識が必要

Copilot Analystエージェントを使う方法

Excelファイルを添付する



自然な言葉で指示する
(例：『合計を出して』『データを分析して』)



Analystエージェント (AI) が処理



集計結果・グラフ・傾向の説明が
自動で出てくる

Analystエージェントは Microsoft 365 Copilot の中で使えるデータ分析機能

⑤ AI は最新の知識を持っているわけではない



対話型AIの知識は学習データの時点で止まっている — 最新情報にはグラウンディングが必要

対話型AIの知識には
時間的な限界がある

知識の
カットオフ

学習データの
期間

未知の期間
(知識なし)

現在

未知の期間の出来事はAIは知らない
→ 誤った情報を自信満々に生成する
(ハルシネーション)

対策

グラウンディング
(外部情報で補強する)

実例

参照データを
渡す
(最新資料をAIに
読み込ませる)

Web検索を
指示する
(最新のWebページ
を調査させる)

Microsoft 365
Copilotの
Researcher
エージェント

Researcherエージェント

Web
ページ
A

Web
ページ
B

Web
ページ
C

根拠のある回答を
ユーザーへ

グラウンディングでもAIの回答は確率的生成のため正しさは保証されない。最終確認は人間の役割

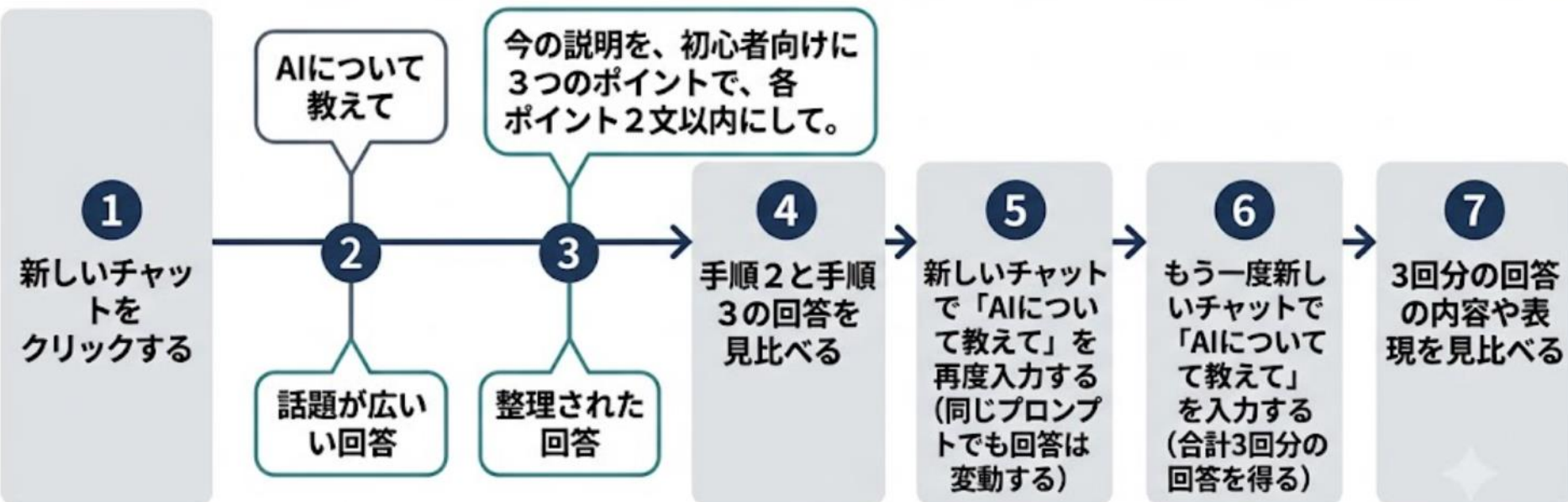
演習

演習① Copilot で対話と生成のしくみを体験する

生成の仕組み — 同じ質問でも
回答が毎回同じではない

対話としての定義 —
前の回答を受けて会話が続く

プロンプト — 指示を
具体的にすると回答が整う



個人情報や機密情報は入力しない

演習② Copilot と協働して告知文を段階的に改善する



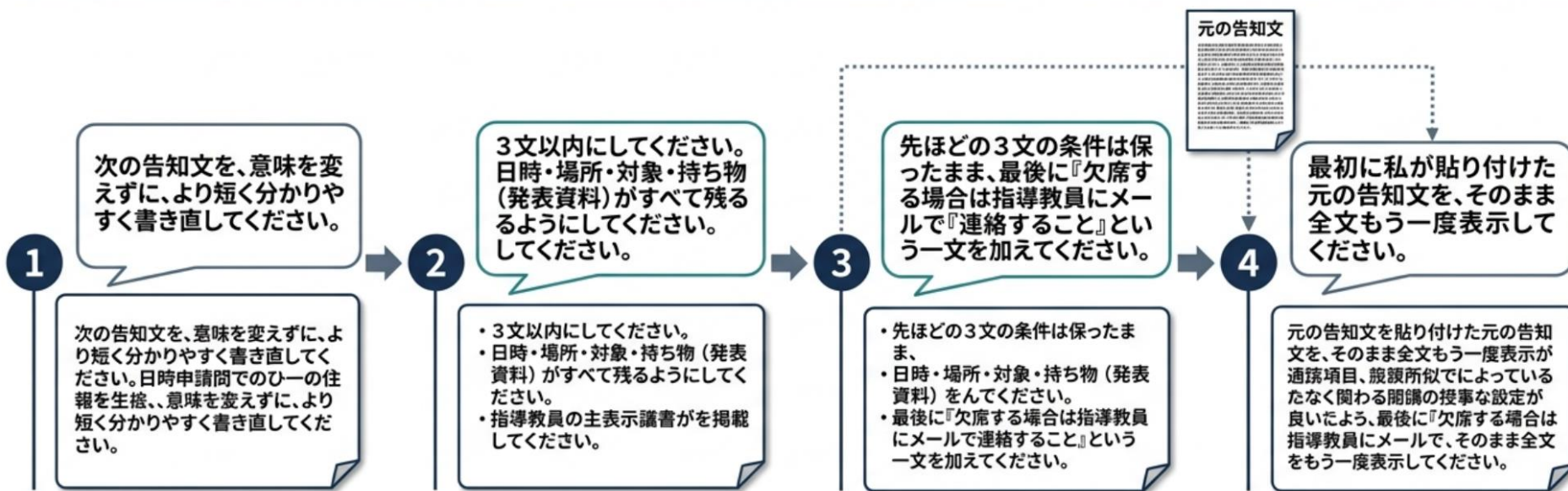
反復的対話 — 修正指示を重ねて文章を改善する



後の指示が前のやり取りを踏まえる



コンテキスト — 最初に与えた文章を後から参照できる



ヒント



修正の指示は、前回までの回答を踏まえて一つずつ積み重ねるように送る／出力の最終的な採否と、その結果への責任は利用者が負う

演習③ Copilot で表形式データを活用する

マルチモーダル — 表の入力とグラフ画像の出力を扱える

データ活用 — 集計結果を元データと照合する

1

表データ (見出し行「タイトル, 分類, 発行年, ページ数」を含む8冊分) をコピーして入力欄に貼り付ける

タイトル	分類	発行年	ページ数

2

分類ごとの冊数を数えてください。

- 教科書: 3
- 専門書: 3
- 一般書: 2

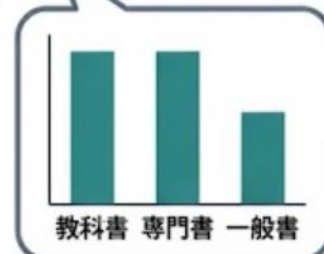
3

回答に示された冊数を、貼り付けたデータを自分でも数えて、一致するかを照合する



4

この分類ごとの冊数を、棒グラフの画像にしてください。



5

文字ではなくグラフの画像が返ることを確認して、その内容が手順3で確認した冊数と合っているかを確認する



ヒント

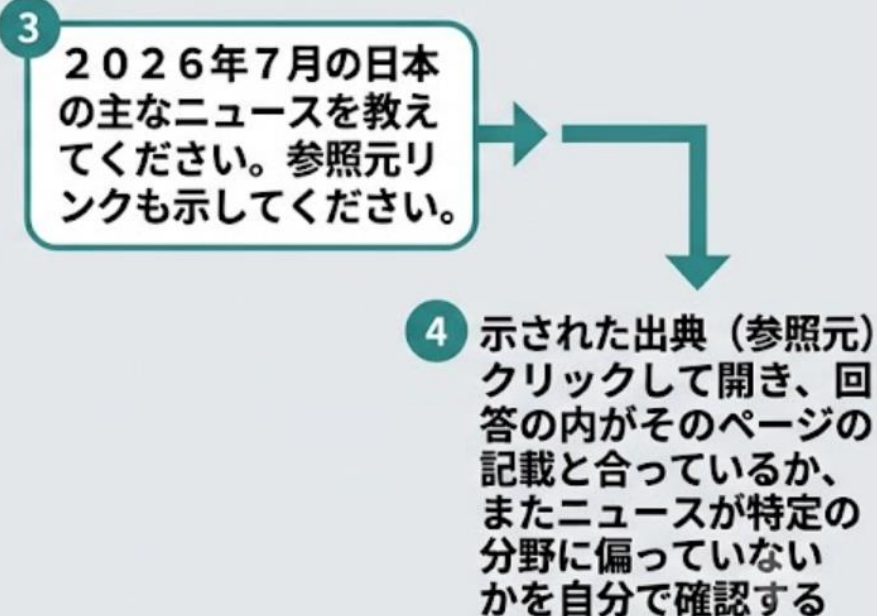
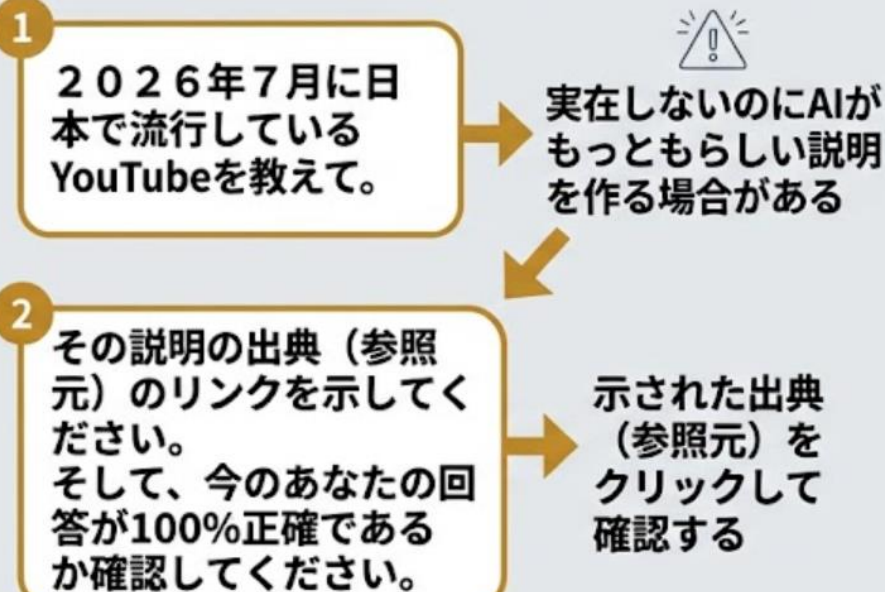
データは見出し行 (タイトル、分類、発行年、ページ数) を含めて貼り付ける

演習④ Copilot を知識ベースとして使い、誤りを見抜く

目的：ハルシネーション
— もっともらしい誤答が実際に起こる

検証 —
出典による検証で誤りに気づける

グラウンディング —
正しい主題では出典がWeb検索で補われる



出典を求めることが、回答の誤りに気づく手がかりになる／AIを情報探索のツールとして使う