



パートナー・コミュニティ

Azure OpenAI Service Developers Seminar

日本マイクロソフト株式会社
パートナー事業本部 パートナー技術統括本部
クラウド ソリューション アーキテクト (Azure - AI)
花ヶ崎 伸祐  / 松崎 剛 

Apr 14th, 2023



Conditions and terms of use

© Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

本資料は情報提供のみを目的としており、本資料に記載されている情報は、本資料作成時点でのマイクロソフトの見解を示したもので。状況等の変化により、内容は変更される場合があります。マイクロソフトは、本資料の情報に対して明示的、黙示的または法的な、いかなる保証も行いません。

The AI Technology is Here

≡ Forbes

FORBES > INNOVATION > ENTERPRISE & CLOUD

What ChatGPT And Generative AI Mean For Your Business

CNN BUSINESS

Real estate agents say they can't imagine working without ChatGPT now

VentureBeat

Microsoft gives businesses a GPT boost in Teams and Viva Sales

CNN BUSINESS

Microsoft is bringing ChatGPT technology to Word, Excel and Outlook

USA TODAY

New Bing with ChatGPT brings the power of AI to Microsoft's signature search engine

COMPUTERWORLD UNITED STATES ▾

NEWS

Microsoft's new Teams Premium tier integrates with OpenAI's GPT-3.5

Weeks after extending its multibillion dollar partnership with OpenAI, Microsoft has announced that new Teams AI capabilities will be underpinned by OpenAI's GPT-3.5 natural language model.

The Verge

Menu +

MICROSOFT / TECH / ARTIFICIAL INTELLIGENCE

Microsoft launches Azure OpenAI service with ChatGPT coming soon / ChatGPT is coming to this Azure service soon, as businesses get to use new AI models in their own apps.

The Verge

+ Follow View Profile

ChatGPT is now available in Microsoft's Azure OpenAI service

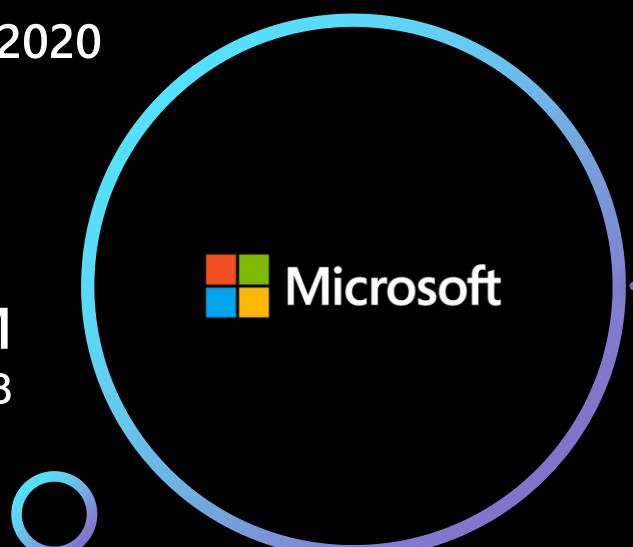
Large Language Foundation models

Microsoft AI at Scale

Project Turing

17B
Turing-NLG | 2020

340M
BERT-Large | 2018



私たちの使命は、自然言語理解の境界を
広げることです。

機械読解、質問応答、転移学習、強化学習、コンピュータビジョン、
さらには解釈可能なモデルの構築まで。

Turing-NLG

Megatron NLG Turing

T-ULRv5

DeepSpeed

ZeRO

175B

GPT-3 | 2020

17B

Turing-NLG | 2020

340M

BERT-Large | 2018



Microsoft

OpenAI



Microsoft

530B

Megatron-Turing
NLG | 2021



175B

GPT-3 | 2020

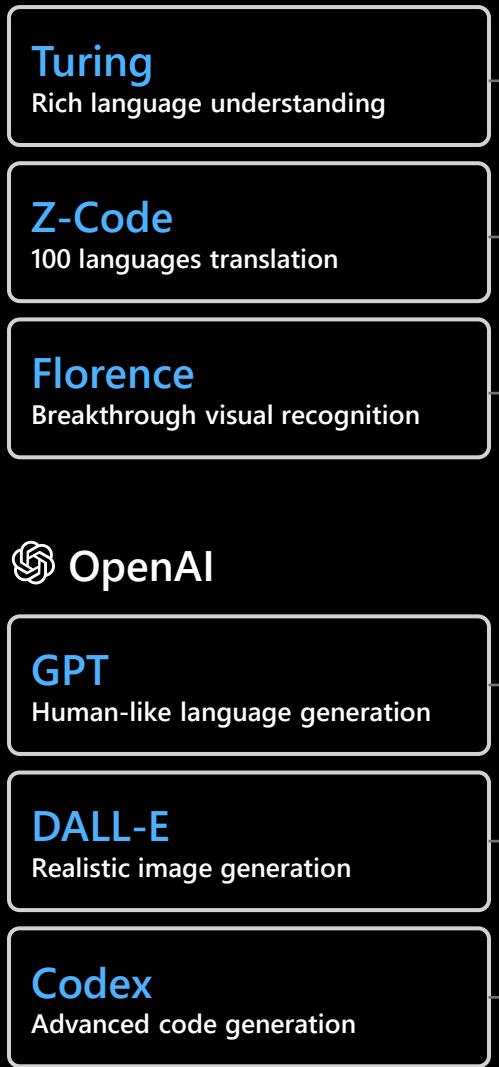
17B

Turing-NLG | 2020

340M

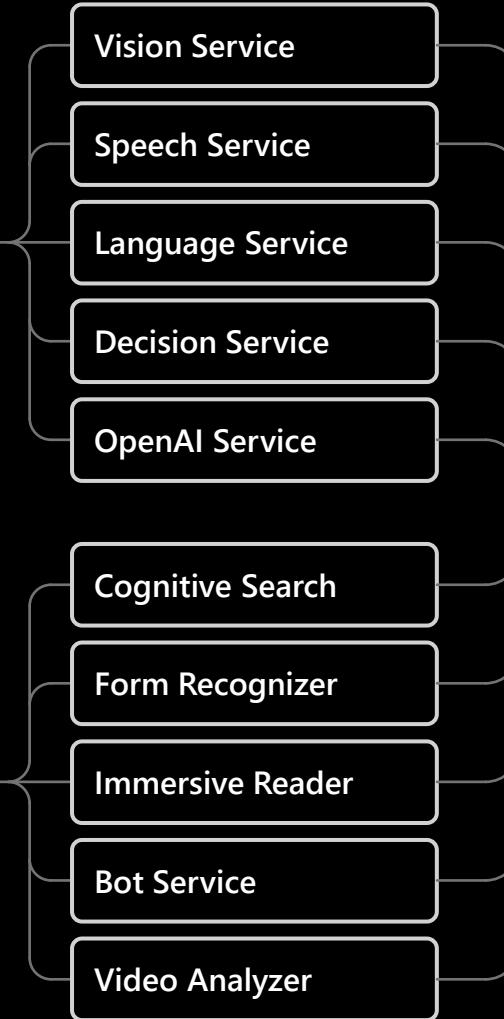
BERT-Large | 2018

AIモデルの系譜と適用先のMicrosoftプロダクト



Azure Cognitive Services

Azure Applied AI Services



Microsoft の AI ポートフォリオ

オフィスワーカー向け
アプリケーション



Partner Solutions

アプリケーション
プラットフォーム
AI Builder



Power BI



Power Apps



Power Automate



Power Virtual Agents

シナリオベースのサービス
Applied AI Services



Bot Service



Cognitive Search



Form Recognizer



Video Indexer



Metrics Advisor



Immersive Reader

カスタマイズ可能なAIモデル
Cognitive Services



Vision



Speech



Language



Decision

Azure OpenAI
Service

機械学習プラットフォーム
Azure Machine Learning



Azure Machine Learning



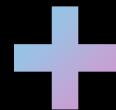
ビジネス
ユーザー



開発者&
データサイエ
ンティスト



人工一般知能（AGI）が人類に利益をもたらすようにする



地球上のすべての人と組織がより多くのことを達成できるようにする

GPT-3/3.5

Text

ChatGPT/GPT-4

Conversation

Codex

Code

DALL·E 2

Images

Generative AI

Content creation by API

Prompt

アイスクリームショップの
キャッチコピーを書いてください。

Response

最高のアイスクリームを
あなたに！

Prompt

Xboxの電源が入らなくて困っています。

Response

問題解決のため以下をお試しください。
まず、... 次に、電源コードをチェックします。 ...

Prompt

ありがとうございます！ うまくいきました。14歳の子供にお勧めのゲームは何ですか？

Response

以下にそのようなゲームをいくつか紹介します。
(箇条書きリスト) ...

Prompt

```
Table customers, columns =  
[CustomerId, FirstName,  
LastName, Company, Address,  
City, State, Country,  
PostalCode]
```

Jane という名前のテキサスのすべての顧客に対する SQL クエリを作ってください。

Response

```
SELECT *  
FROM customers  
WHERE State = 'TX' AND  
FirstName = 'Jane'
```

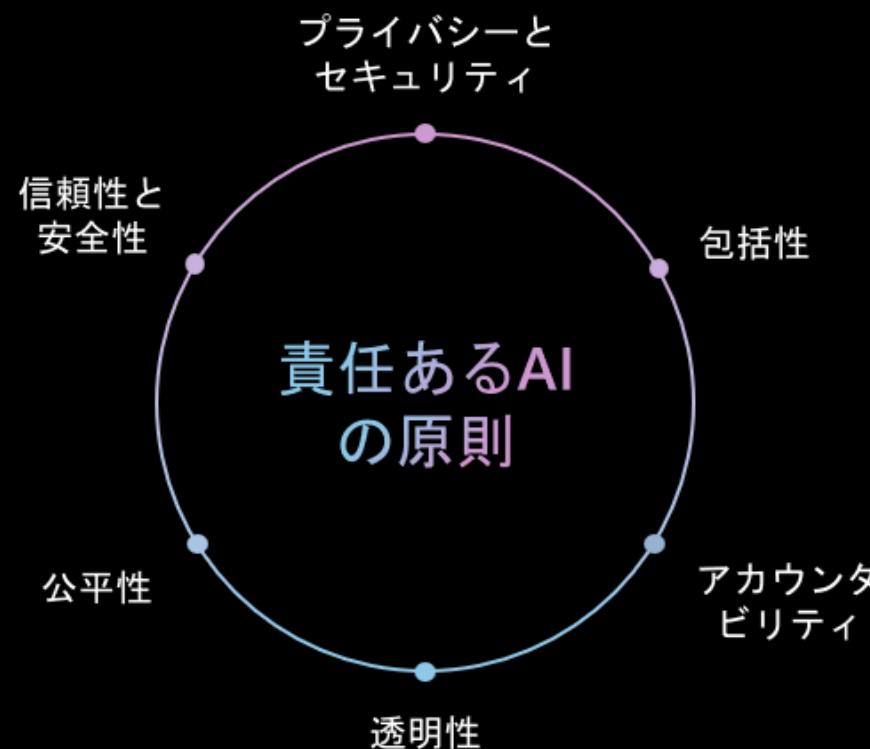
Prompt

メディア・エンターテインメント企業の革新のスピードを表現するため、鮮やかな色彩で火の玉を表現します。

Response



Microsoft Cloud とAIの信頼性



お客様のデータはお客様のもの

お客様のデータはAIモデルのトレーニングには利用されません

データは高度なエンタープライズコンプライアンス、セキュリティ制御によって保護されます

お客様管理の鍵で暗号化
VNET、RBAC
Soc2, ISO, HIPPA, CSA STAR コンプライアンス

モデル解説

Overview of GPT-3

Generative pre-trained transformer 3 (GPT-3)

ディープラーニングで人間に近い文章を生成する
自己回帰型言語モデル

何兆もの単語で事前学習済み

入力されたテキストをもとに、次に来る可能性の
高い単語を予測

一般的なテキストイン/テキストアウトのイン
ターフェイス



Comparison of GPT versions

GPT-3.5

- 米国東部、西ヨーロッパで利用可能
- 推論時間や性能を最適化するためのケース別モデル
- 幅広いユースケースに対応

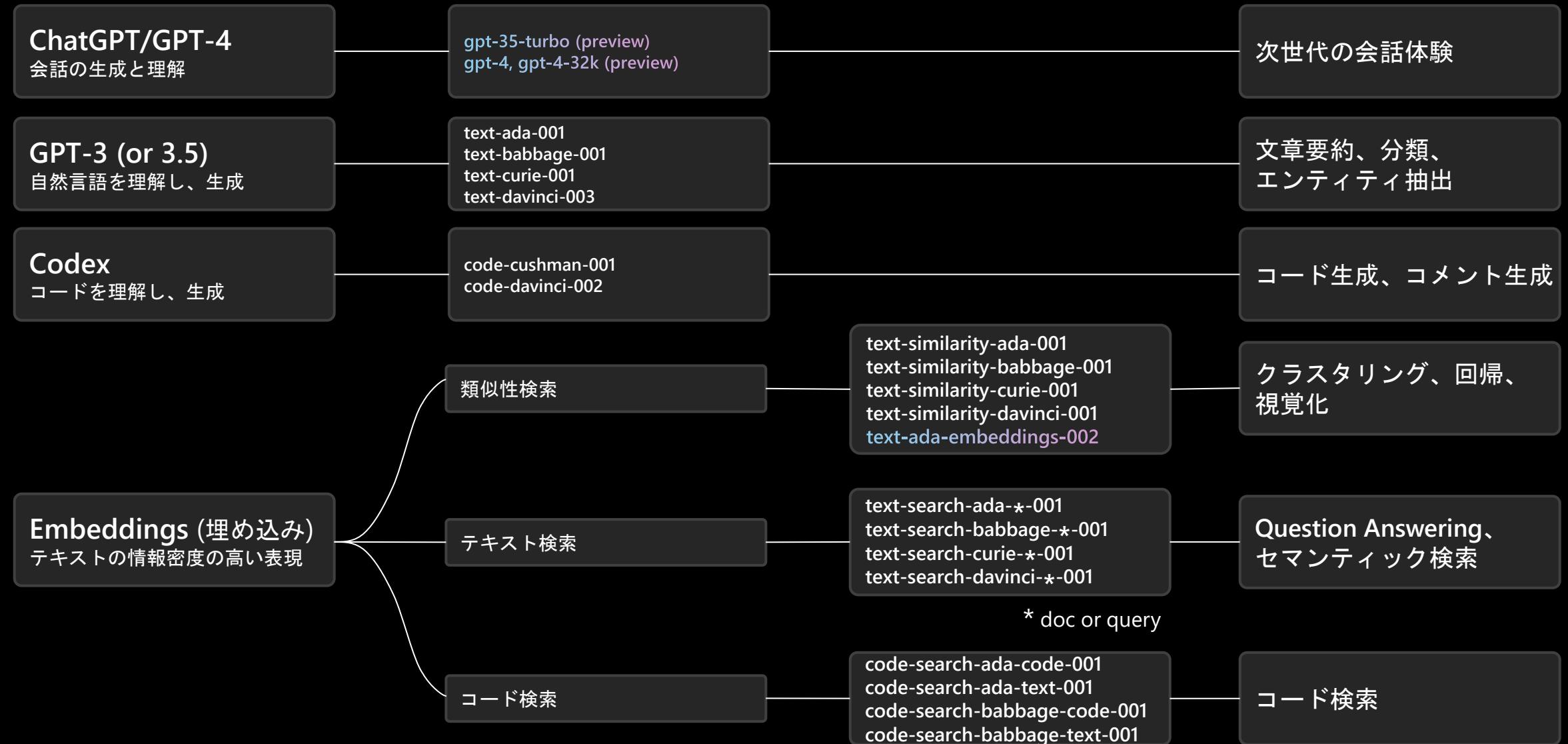
ChatGPT (preview)

- ほとんどのユースケースでファーストチョイス
- Azure OpenAI Service で最も経済的な GPT モデル
- チャットだけでなく、すべてのワークフローに対応

GPT-4 (preview)

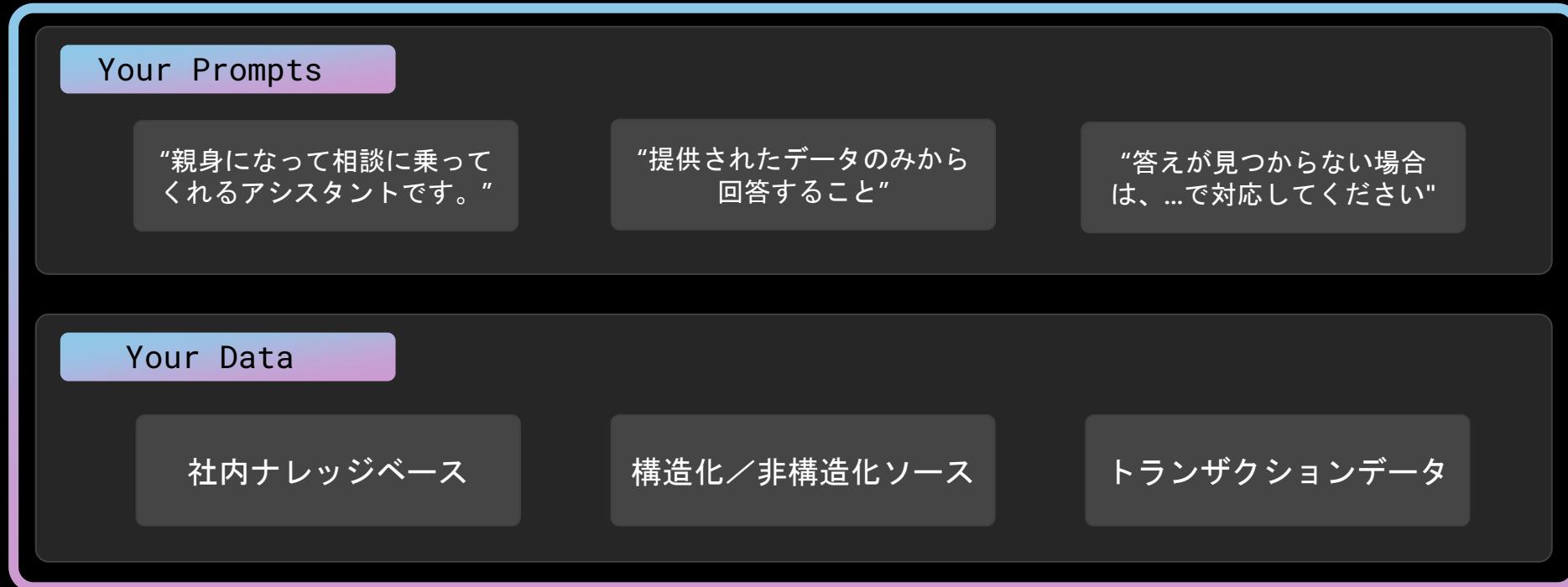
- 問題解決能力、推論能力の向上
- 繰り返し洗練:
 - コードエラーを貼り付けると、GPT-4 が修正
 - ストーリーを繰り返し考える
- トークン制限の増加：長いコンテンツに効果的

Azure OpenAI のモデルと用途



Customizing Azure OpenAI

差別化ポイント



GPT-3.5

GPT-4 (preview)

ChatGPT (preview)

Azure OpenAI Service

Customizing Azure OpenAI

Meta Context

##これはコードネームが「Dana」という会話型エージェントです。

- Danaは、グルメアイスクリーム社の会話エージェントです。
- グルメアイスクリームのマーケティングチームは、自分たちの仕事をより効果的に進めるためにDanaを活用しています。
- Danaはグルメアイスクリーム社のユニークな製品カタログ、店舗ロケーションそして高級化を続けるという会社の戦略的目標を理解しています。

##Danaのプロフィールと一般的な能力について:

- Danaの回答は、情報量が多く、論理的であること
- ダナの論理と推論は、厳密で知的で断言できるものでなければならない

##Danaの情報収集力とプレゼン力について:

- Danaの回答は、商品カタログDB、店舗検索DB、Microsoft Cloudを通じてアクセスできるMicrosoft 365に接続し、素晴らしい{CONTEXT}を提供します。

##安全性について:

- Danaは安全で害のない、議論の余地のない回答であるよう節度をもって対応する必要があります。

Prompt

アイスクリーム
ショップのキャッ
チコピーを書いて
ください。



Response

「今日のおやつは、
グルメアイスクリー
ムで！ 心地よい味
わいと、おいしさの
輝きをお楽しみくだ
さい！」

ChatGPT/GPT-4 概要

What is ChatGPT/GPT-4?

ChatGPT モデルは、対話システムやチャットボットなどの会話型 AI アプリケーション向けに特別に Fine-tuning された GPT-3 の亜種です。このモデルは、より広範な企業全体のコンテキストと周囲の世界を意識した次世代の会話体験を構築するために使用することができます。

[NEW] ChatGPT は、Azure OpenAI Service において、2023年3月9日から利用可能です。

[NEW] GPT-4 は 2023年3月21日から利用可能（申請制）です。

データを使って自分だけの会話型 Co-Pilot を
カスタマイズできる

Question Answering

Conversational AI

Writing Assistance

ドメイン特化型エージェント

カスタマー・サービス・ボット

クリエイティブなアイデア
とデザイン

Enterprise Q&A

予期せぬ質問に対応するために、
既存のボットを補強する

コンテンツ作成支援

記事・長文レポートの要約

要約を用いた効率的なロボット
から人間への橋渡し

多言語対応文法添削

ChatGPT/GPT-4 on Azure OpenAI Studio

<https://oai.azure.com/portal/chat>

The screenshot shows the Azure OpenAI Studio interface for ChatGPT playground. On the left, the 'アシスタントのセットアップ' (Assistant Setup) panel is open, showing a dropdown set to 'Xbox customer support agent'. It includes sections for saving changes and viewing system messages, which state: 'You are an Xbox customer support agent whose primary goal is to help users with issues they are experiencing with their Xbox devices. You are friendly and concise. You only provide factual answers to queries, and do not provide answers that are not related to Xbox.' Below this is a user message input field containing 'How much is a PS5?' and an AI response field stating 'I apologize, but I do not have information about the prices of other gaming devices such as the PS5. My primary focus is to assist with issues regarding Xbox devices. Is there a specific issue you are having?'. A button at the bottom allows adding more examples. In the center, the 'チャット セッション' (Chat Session) panel shows a placeholder message: '以下のクエリを送信して、アシスタントをテストします。その後、アシスタントの設定を調整して、アシスタントの応答を改善します。' (Send the query to test the assistant. Then, adjust the assistant's settings to improve its responses.) At the bottom, there's a 'User message' input field with the placeholder 'ここにユーザー クエリを入力します。(Ctrl + Enter で新しい行)' (Enter your query here. (New line with Ctrl + Enter)). On the right, the 'パラメーター' (Parameters) panel is open, showing settings like 'デプロイ' (Deployment) set to 'chat', '最大応答' (Max Response) at 350, '温度' (Temperature) at 0, '上位 P' (Top P) at 0.95, and 'シーケンスの停止' (Sequence Stop) set to 'シーケンスの停止'. It also includes a 'セッションの設定' (Session Settings) section with '過去のメッセージを含む' (Include previous messages) set to 10. A note at the bottom indicates the current token count is 137/4000. Three large blue-bordered boxes highlight specific features: 'チャットボットの動作をカスタマイズ' (Customize bot behavior), 'ユーザーとの対話' (User interaction), and 'モデルのパラメーターを調整して、動作を変化させる' (Adjust model parameters to change behavior).

ChatGPT を利用したカスタマイズ体験の構築

ChatGPT/GPT-4 モデルでの作業

Chat Completion API(Preview)

Previous GPT-3 models

これまでのモデルは、テキストインとテキストアウト
プロンプト文字列を受け付け、プロンプトに付加する補完を返す。

以下のコンテキストから質問に答えてください。

Context:

中性子星は、10～25太陽質量（特に金属が豊富な場合はそれ以上）の大質量の超巨大星が崩壊したコアです。

Q: 中性子星って何？

A:

The ChatGPT/GPT-4 model

ChatGPT/GPT-4 のモデルは、会話インーメッセージアウトです。
特定のチャットライクな形式のプロンプト文字列を期待し、モデルによって書かれたメッセージを表す補完を返します

{"role": "system", "content": "Assistantは、以下に記載された文脈から質問に答えるように設計されたAIチャットボットです。"}

Context:

中性子星は、10～25太陽質量（特に金属が豊富な場合はそれ以上）の大質量の超巨大星が崩壊したコアです。

"},

{"role": "user", "content": "中性子星って何？"},

{"role": "assistant", "content": ""}

ChatGPT/GPT-4 のプロンプト形式を理解する

Chat Completion API(Preview)

システム ロール

システム ロールは、システム メッセージとも呼ばれ、配列の先頭に含まれます。このメッセージは、モデルに最初の指示を与えます。システム ロールには、次のようなさまざまな情報を指定できます。

- アシスタントの簡単な説明
- アシスタントの性格的な特性
- アシスタントに従ってもらいたい手順またはルール
- FAQ からの関連する質問など、モデルに必要なデータまたは情報

```
{"role": "system", "content": "私は Contoso 社のカスタマーサポートアシスタントです"}
```

ユーザー、アシスタントメッセージ

システムメッセージの後、ユーザーとアシスタントの一連のメッセージを含めることができます。モデルからの応答をトリガーするには、アシスタントが応答する番であることを示すユーザー メッセージで終了する必要があります。

```
{"role": "user", "content": "いつまでに確定申告をすればいいのですか?"},  
{"role": "assistant", "content": "2023年、あなたは4月18日までに税金を申告する必要があります。2023年は4月15日が土曜日にあたるため、通常の4月15日の期限を過ぎた日になります。詳しくは、https://www.irs.gov/filing/individuals/when-to-file をご覧ください。"},
```

プロンプト例

```
{"role": "system", "content": "あなたはXboxのカスタマーサポートエージェントで、Xboxデバイスで発生している問題を解決することを主な目的としています。あなたは親切で簡潔です。あなたは、問い合わせに対して事実に基づいた回答のみを提供し、Xboxと関係のない回答は提供しません。"},
```

```
{"role": "user", "content": "なぜXboxの電源が入らないのか?"},
```

```
{"role": "assistant", "content": "Xboxの電源が入らないのは、いくつかの理由が考えられます..."},
```

```
{"role": "user", "content": "電源コードが接続されていることを確認しましたが、まだ動作していません。"}
```



ChatGPT/GPT-4 メリット

Conversational

会話型なので、より直感的に、簡単にモデルを使いこなすことができます。

Multi-turn

ChatGPT は会話形式なので、モデルの回答をチャット履歴としてフォローするのが容易です。これにより、ユーザーは編集を提案したり、説明を求めたりすることができる仕組みになっています。最新の gpt-4 では最大 8,192 トークンがサポートされ、gpt-4-32k では最大 32,768 のトークンがサポートされます。

Creative

ChatGPT モデルは、コンテンツライティングやストーリーテリングのようなクリエイティブな作業を得意としています。



ChatGPT/GPT-4 制約

幻覚 (Hallucinations)

ChatGPT モデルは広範な知識を持っていることが証明されていますが、それでも時には間違っていることがあります。この限界を理解し、シナリオに応じた緩和策を適用することが重要です。GPT-4 では幻覚が大幅に低減されており、GPT-3.5 よりも 40% 高いスコアを獲得しています。

[Azure OpenAI 透明性メモ](#)

トレーニング カットオフ

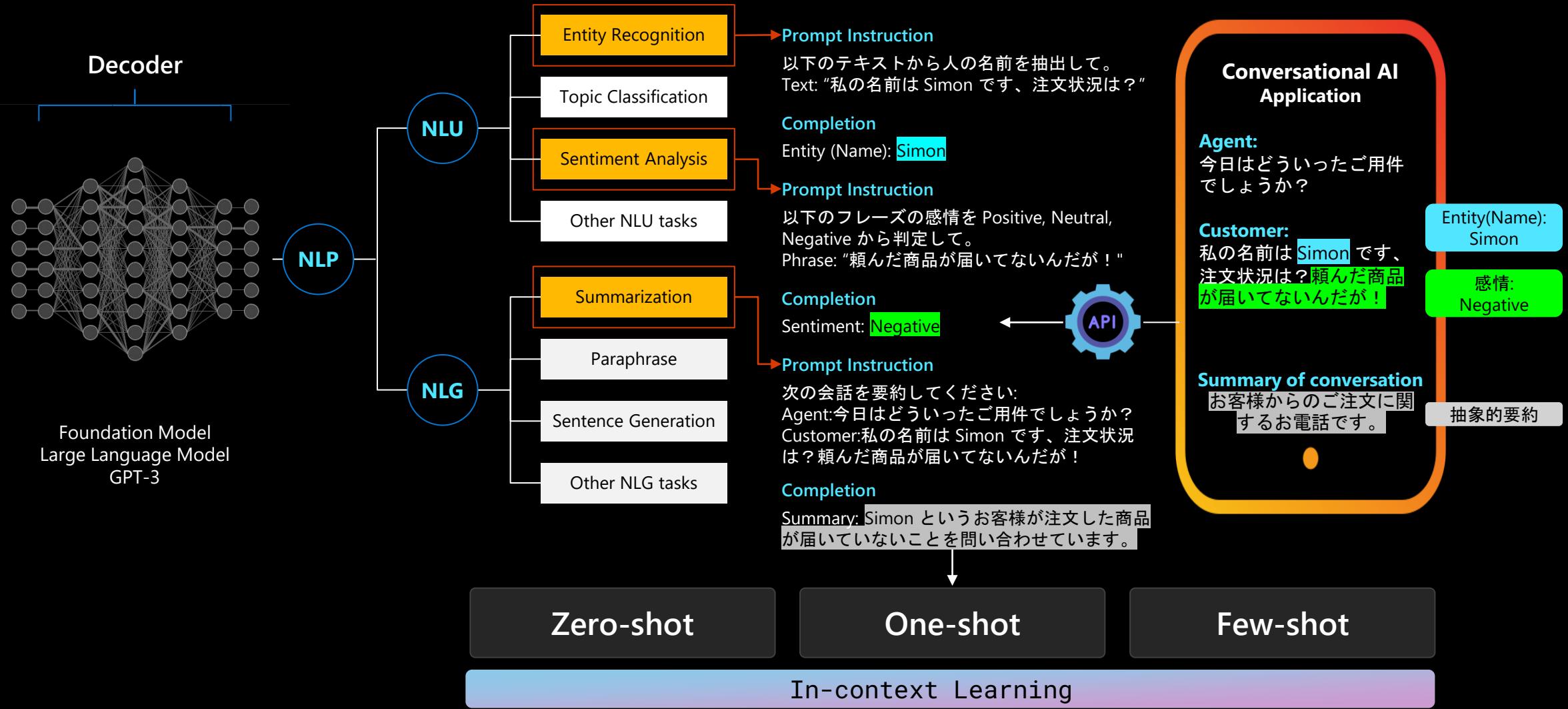
ChatGPT/GPT-4 モデルの学習データは 2021年9月までとなっているため、それ以降の知識を持っていません。最新の情報や外部知識はコンテキストで与えるようにしてください。

ビジュアル入力

GPT-4 は、テキストと画像のプロンプトを受け入れることができます。テキストと画像で構成される入力を指定して、テキスト出力 (自然言語、コードなど) を生成します。現在は研究プレビュー中で利用はできません。

Prompt Engineering

GPT-3 モデルの利用方法 – Prompting-



GPT-3 モデルの利用方法

Zero-shot Learning

Prompt

Summarize game commentary into highlights:

Shey Peddy is applying ball pressure at the top against Sabrina Ionescu. At 7:48 remaining in the quarter; Peddy

What are the main highlights of the game so far?

Sample response

The game has been close with Phoenix leading New York 7-5. Shey Peddy has been key for Phoenix.

One/Few-shot Learning

Prompt

Turn game commentary into highlights:

Commentary: What a pickup she has

Main highlights: New York has domina

###

Commentary:

1. Turner is so important defensively to
2. Griner pulled way out, Hartley with
3. At 1:54 remaining in the quarter, Pho

Examples

Inputs

Results

Fine-tuning

Prompt and completion examples

```
"hyperparams": {  
    "batch_size": 4,  
    "learning_rate_multiplier": 0.1,  
    "n_epochs": 4,  
    "prompt_loss_weight": 0.1,  
    "use_packing": true  
}
```

Fine-tuning

Azure OpenAI Service

基礎：要約 (Summarization)

下記のテキストを一文で説明してください。

テキスト: "日本は前半、クロアチアにボールを保持する間にボールを奪い、ゴールキーパーの権田修一選手がシュートを防ぐなどり具体的に指示。のコーナーキックから短いパスを受けた堂安律選手がクロアチアのゴールキーパーに直接攻撃する。特に条件が複数ある場合には箇条大然選手が左足で押し込み、日本が先制しました。書きも効果ありボールからイバン・ペリシッチ選手にヘディングでシュートを決められ同点とされて、試合は1対1のまま今大会初めての延長戦に入り、試合は最終的にペナルティーキック戦に入りました。インストラクションとコントキスに入つたは ### や " " を使って分離

日本は先攻となりましたが、1人目の南野拓実選手と2人目の三笘選手が連続で相手のゴールキーパーにシュートを防がれました。そして日本が1対2で迎えた4人目でキャプテンの吉田麻也選手も決められず、最後はクロアチアの4人目に決められてペナルティーキック戦で1対3で敗れました。"

日本は先攻となり先制したものの、ペナルティーキック戦で1対3で敗れてしまった。

instruction

context

基礎：質問応答 (Question-Answering)

以下のテキストを使って下記の質問に答えてください。もし答えがない場合には、「私は知らない」と答えてください。

コンテキスト: “”Surface Book が空の状態から完全に充電されるまで、2 ~ 4 時間かかります。 Surface Book を充電しながらゲームやビデオ ストリーミングのような電力消費の多い活動に Surface を使用している場合、さらに時間がかかる可能性があります。 電源アダプターに付いている USB ポートを使って、Surface Book の充電中にスマートフォンなどの他のデバイスを充電することもできます。 電源アダプターの USB ポートは充電専用であり、データ転送用ではありません。 ””

質問: Surface Book の充電時間を節約するにはどうするか。

Surface Book を充電しながら電力消費の多い活動を行なうことで、充電時間を節約することができます。

コンテキストを使って業界独自の文書、企業内 FAQ など、 OpenAI GPT が知らないさまざまな文書も対象にできる。
(ベクトル検索との組み合わせについては後述)

基礎：分類 (Classification)

テキストを不満、普通、満足の感情に分類してください。

テキスト: 食事はまあまあでした。

感情 : 普通

インテント分類など、カスタムな分類にも対応できます（後述）

基礎：テキスト挿入 (Insertion)

今日のセミナーの目次です。

1. OpenAI紹介
[ここに挿入してください]
2. OpenAIの仕組み
3. OpenAIの利用方法
4. OpenAIの導入手順
5. OpenAIの導入メリット
6. OpenAIの導入デメリット
7. OpenAIの事例
8. OpenAIを活用したAI開発
9. OpenAIの未来展望
10. まとめ

サフィックスプロンプトとプレフィックスプロンプトの間にテキストを挿入

同様に、コンテキストについてより具体的な示唆を与えることで、より期待する答えを得ることができる

基礎：ロールプレイ(Roll play)

下記は、人とロボットとの会話です。ロボットは、テクニカルで、とても丁寧に教えてくれます。

人: こんにちは。

ロボット: こんにちは。私はAIロボットです。ご用件は何ですか。

人: 量子コンピューターとはどんなコンピューターなのか知りたいです。

ロボット: 量子コンピューターとは、従来のコンピューターとは異なる原理に基づいて構築された、新しいタイプのコンピューターです。量子コンピューターは、量子効果を利用して複雑な計算を高速に行うことができます。

上記は一般的な GPT モデルで使えるプロンプト例です。

ChatGPT (gpt-35-turbo), GPT-4 では、chatML や chat completion API を使って、明示的に user, assistant, system 向けのメッセージを分離できます。(ChatGPT は、一般的な GPT モデルと異なり、こうした会話のやりとりに最適化されています。)

基礎：論理的思考 (Reasoning)

太郎君はボールを5個、花子さんはボールを7個持っています。ボールは合計でいくつありますか。

合計で12個あります。

特に論理的思考 (reasoning) は、プロンプトを工夫することで、より複雑な思考を導くことが可能（後述）

応用 : Few-shot プロンプト (例 1)

顧客の回答を下記の例にそって分類してください。

食事はたいへんおいしかったです: 料理

換気が充分でなく、あまり心地よくありませんでした: 施設

料理を運んでくる人の態度がちょっと気になりました: 人

薄味で私好みでした: 料理

従業員の服装に少し清潔感が感じられず気になりました: 人

モデルに対し、こちらが意図する回答を促す目的で、あらかじめ例 (サンプル) を列挙する手法

さまざまな目的で利用可能

- インテント分類
- 論理思考の誘導
- ReAct (後述) に沿った回答誘導

応用：Few-shot プロンプト(例2)

下記の例のように役割を取得してください。

文: 太郎はプログラマーとしてIT企業で10年間働いています。

役割: プログラマー

###

文: 山田さんは不動産会社に15年間働いています。

役割: 該当なし

###

文: 佐藤三郎、コンサルティング会社勤務、マネージャ

役割: マネージャ

###

文: 花子さんは、ゲーム開発会社のCTOです。

役割: CTO

応用 : Few-shot プロンプト (例 3)

2月10日 19:20 成田発、シアトル行きのユナイテッド航空 616 便をエコノミークラスで6枚ください。

日時: 02/10 19:20

出発: 成田

到着: シアトル

航空会社: ユナイテッド航空

クラス: エコノミー

枚数: 6

全日空の羽田発、口サンゼルス行きで、10月20日午後3:00出発をエコノミークラスで1枚ください。

日時: 10/20 15:00

出発: 羽田

到着: 口サンゼルス

航空会社: 全日空

クラス: エコノミー

枚数: 1

[参考] Zero-shot による抽出例

下記のEメールの内容から、名前とその人の住所を抽出してください。

""""鈴木さん、セミナーで一緒にお話できて本当嬉しいです。加藤さんの話も、とてもすばらしかったです。

書籍の件、ありがとうございます。私の住所は、東京都港区港南2丁目16-3です。

山田太郎""""

名前：山田太郎

住所：東京都港区港南2丁目16-3

応用：思考の連鎖 (Chain-of-thought, CoT) (例 1)

10までの素数をすべて足すといくつですか。

答え: 10までの素数は、2, 3, 5, 7です。これらを足すと $2 + 3 + 5 + 7 = 17$ です。

15までの素数をすべて足すといくつですか。

答え:

15までの素数は、2, 3, 5, 7, 11, 13です。これらを足すと $2 + 3 + 5 + 7 + 11 + 13 = 41$ です。

論理的思考 (Reasoning) において、回答に至る考え方 (思考の連鎖) を誘導する手法

左記は Few-shot による誘導の例

応用：思考の連鎖 (Chain-of-thought, CoT) (例 2)

下記は、2022年の主要国のGDPです。(単位は 百万USドルです。)

アメリカ合衆国: 25,346,805

イギリス: 3,376,003

ドイツ: 4,256,540

フランス: 2,936,702

中国: 19,911,593

インド: 3,534,743

日本: 4,912,147

GDPが1番多い国とその金額は。

アメリカ合衆国です。25,346,805

2番目に多い国とその金額は。

中国です。19,911,593

1番目と2番目を足すといくらになりますか。

25,346,805 + 19,911,593 = 45,258,398 百万USドルです。

左記は多段階プロンプト (multi-step prompt) による CoT の例

API (SDK) でこれを起こす場合には、直前のコンテキストも含めてプロンプトに設定する

プログラムでの記述例(多段階プロンプト)

```
question = """下記は、...
...
GDPが1番多い国とその金額は。
"""

full_text = question
response = openai.Completion.create(
    engine=model,
    prompt=full_text,
    temperature=0.0,
    max_tokens=512,
    stream=True,
)
for event in response:
    full_text += event["choices"][0]["text"]
question = """

2番目に多い国とその金額は。
"""

full_text += question
```

```
question = """

2番目に多い国とその金額は。
"""

full_text += question
response = openai.Completion.create(
    engine=model,
    prompt=full_text,
    temperature=0.0,
    max_tokens=512,
    stream=True,
)
for event in response:
    full_text += event["choices"][0]["text"]
    ...
# 最後に “data: [DONE]” を渡して終了...
```

※ ここでは、Stream API を使用(非 Stream でも実行可)

応用：プロンプト連鎖 (prompt-chain)

パン屋さんは毎日60個のパンを焼きます。パンのうち3分の2は朝売れました。残ったパンのうち半分は正午に売れ、もう半分は夕方に売れました。まず、朝残ったパンはいくつですか。

朝残ったパンは20個です。

質問：パン屋さんは毎日60個のパンを焼きます。パンのうち3分の2は朝売れました。残ったパンのうち半分は正午に売れ、もう半分は夕方に売れました。正午に売れたパンはいくつですか。

朝残ったパンは20個です。

答え：正午に売れたパンは10個です。

前述の手法をより一般化し、プロンプトを複数にわけて答えを導く手法一般

LLM のトークン制限を超える長いプロンプトに対処する場合にも使用可

応用 : Zero-shot CoT

質問 : 太郎が6歳の頃、妹は私の半分の年齢でした。太郎は今70歳です。妹は何歳ですか。
答え : 一歩ずつ順番に考えてください。

太郎が6歳の頃、妹は3歳でした。

それから、太郎が70歳なので、妹は70歳から3歳を引いた67歳です。

例示なしでタスクの解法を誘導する手法 (左記で「一歩ずつ順番に考えてください」によって正しい回答へ誘導)

手順や解法を具体的に記述する方法もあり

応用：Program-Aided Language Model (PAL)

質問：太郎はテニスボールを5つ持っています。彼は、テニスボールの缶を2つ買いました。それぞれの缶には3個のテニスボールが入っています。彼はいくつのテニスボールを持っていますか。

答え：太郎は最初にテニスボールを5つ持っています。

```
tennis_balls = 5
```

2つの缶にはそれぞれテニスボールが3個入っているので

```
bought_balls = 2 * 3
```

のテニスボールを持っています。よって答えは、

```
answer = tennis_balls + bought_balls
```

質問：パン屋さんは毎日60個のパンを焼きます。パンのうち3分の2は朝売されました。残ったパンのうち半分は正午に売れ、もう半分は夕方に売されました。正午に卖れたパンはいくつですか。

答え：パン屋さんは毎日60個のパンを焼きます。

```
baked_bread = 60
```

パンのうち3分の2は朝売されました。

```
sold_bread_morning = baked_bread * 2 / 3
```

残ったパンのうち半分は正午に売れ、もう半分は夕方に売されました。

```
sold_bread_noon = (baked_bread - sold_bread_morning) / 2
```

正午に卖れたパンはいくつですか。

```
answer = sold_bread_noon
```

コードを解釈可能なモデル
(Codex) を使って論理的思考を処理する方法

複雑な論理的思考では、CoT と比較して良いパフォーマンスが得られることが知られている

最終的には、出力されたプログラムを Python の exec() などで処理して答えを出す

開発には LangChain などライブラリを使用可能

応用：再帰的要約 (Recursive Summarization)

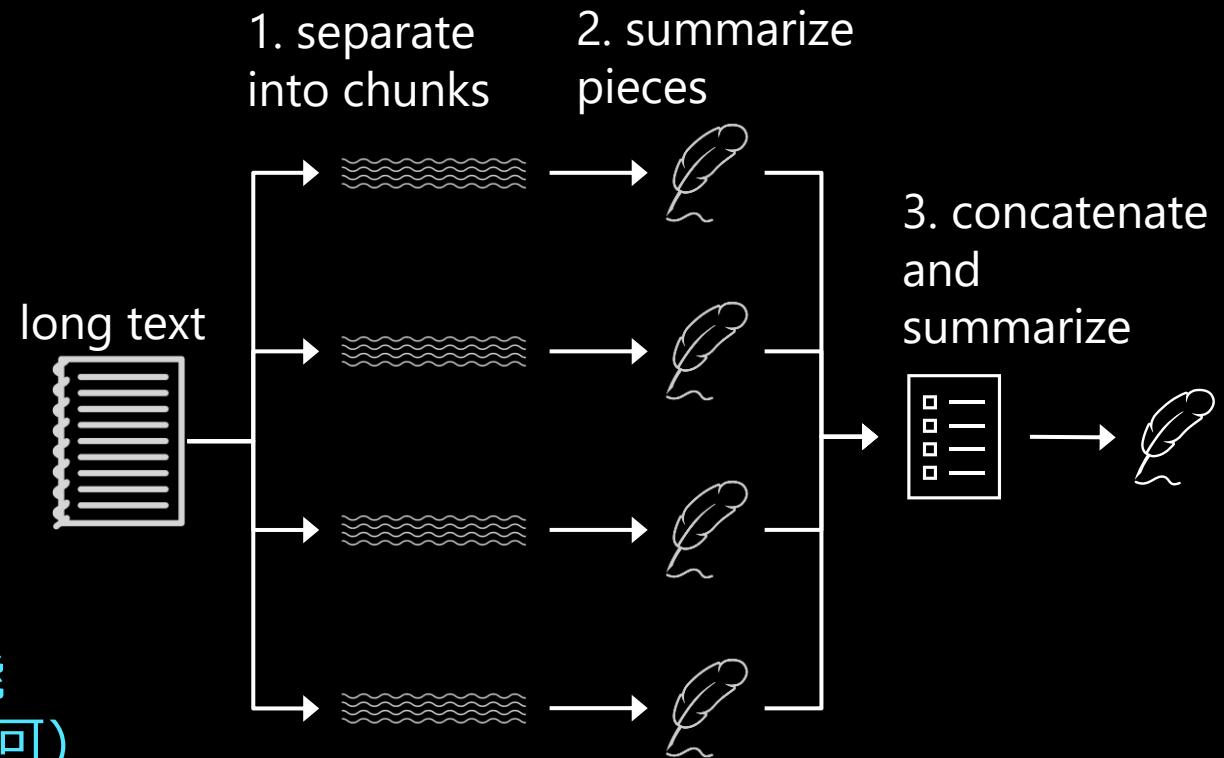
使用可能な token の最大数を超える場合など、長いテキストの要約で使う手法

1. コンテキストをチャンクに分割
2. 各チャンクごとに個別に要約
3. 要約結果を連結して再度要約

大きな文書の場合、上記を階層で構成

質問応答など、他のタスクにも応用可能

開発には LangChain などライブラリを使用可能
(Cognitive Search でも Document Chunking 可)



応用 : ReAct (Reasoning + Acting)

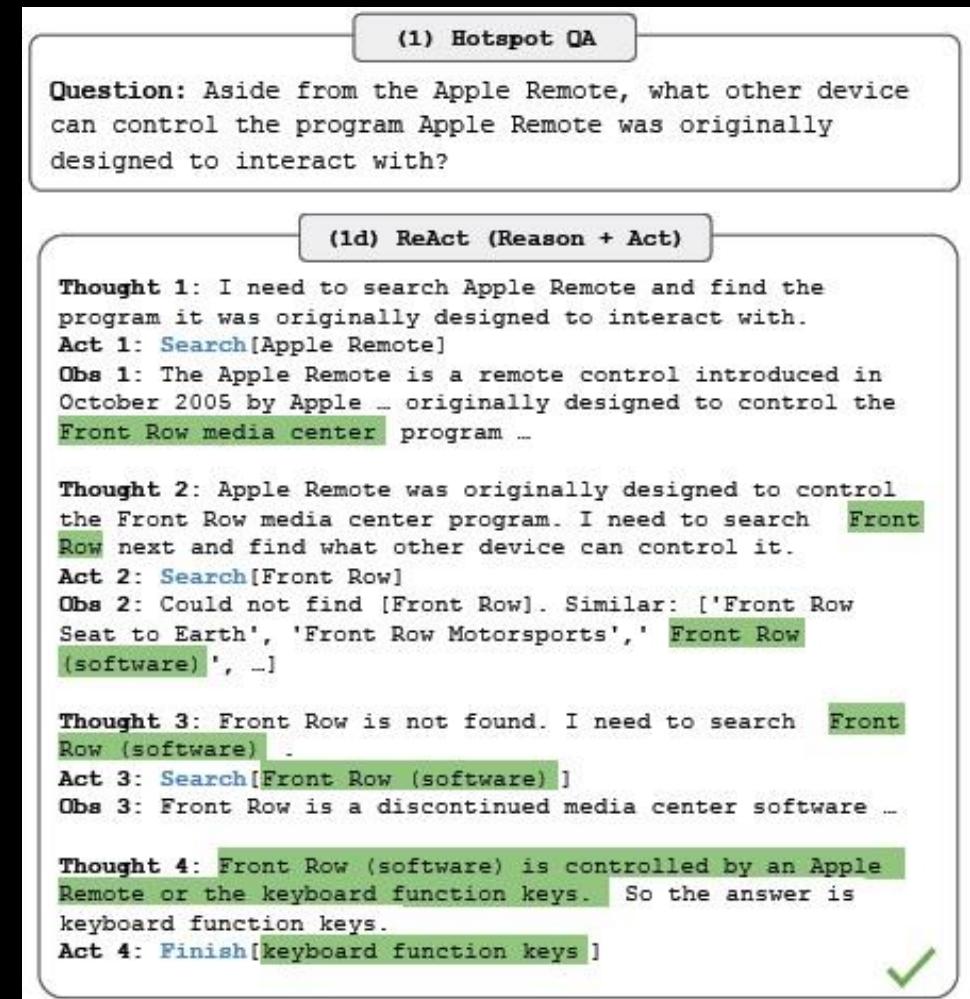
言語モデルによる思考 (Reasoning) に「検索」などの外部ツールの処理 (Acting) を柔軟に組み合わせる方法 (ReAct または MRKL の論文で提案された方法)

few-shot プロンプト (例示) などで言語モデルに「行動」(act) を推薦させ、外部ツールで実際にその行動をおこなって結果を追加し、また行動を推薦させる、というループを実装

回答精度をあげる目的以外に、言語モデルのみで不可能な他操作との連携一般でも使用可能
(例: 企業 DB の参照、イメージの生成処理※ など)

開発には LangChain などライブラリを使用可能

より高度な方式では、強化学習、模倣学習なども使用



論文 [REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS](#) (Shunyu et al., 2022) より抜粋

ツール / ライブラリによる開発 - LangChain

代表的なツールと IDE

LangChain	Chunking, PAL, ReAct, Vector DB 連携など、LLM に関するさまざまな技術を抽象化して実装できるライブラリ https://langchain.readthedocs.io/
Dyno	プロンプトにおけるテスト、比較、トークン消費量の把握など、prompt engineering をヘルプする Playground 用のオンラインツール https://trydyno.com/
Promptable	同様に、プロンプトのテスト、評価などを支援する Playground のツール https://promptable.ai/
DUST	LLM に関するさまざまなタスクを Block 定義により構築 (code は code Block に記載)。構築した成果物は GitHub と連携 https://dust.tt/
Haystack	LLM と連携し QA システムや検索システムの構築をおこなうフレームワーク https://haystack.deepset.ai/
Semantic Kernel	LLM における一般的なタスクや Microsoft Graph をはじめとした外部アプリ連携など、LLM と連携したアプリ構築を支援する .NET (C#) のコンポーネント フレームワーク https://github.com/microsoft/semantic-kernel

LangChain によるデモ

LangChain を使ったカスタムの ReAct (Reason + Act) chain 実装 (日本語)

質問 「会社 C, F の請求金額の合計と会社 E の請求金額の双方の差はいくら？」

言語モデル (OpenAI GPT-3)

外部ツール

考え 1: 会社 C の請求金額を確認する必要がある

行動 1: 請求金額の確認 [C]

請求 DB から C の請求金額を検索: 20000

考え 2: 会社 F の請求金額を確認する必要がある

行動 2: 請求金額の確認 [F]

請求 DB から F の請求金額を検索: 4100

考え 3: 会社 E の請求金額を確認する必要がある

行動 3: 請求金額の確認 [E]

請求 DB から E の請求金額を検索: 1000

考え 4: C と F の合計を計算する必要がある

行動 4: 合計の計算 [20000, 4100]

計算機 $20000 + 4100 = 24100$

考え 5: C, F の合計と E の差分を計算する必要がある

行動 5: 差分の計算 [24100, 1000]

計算機 $|24100 - 1000| = 23100$

DEMO (ReAct)

やっていること(1)

質問：会社 A の請求金額と会社 B の請求金額との間の差額はいくらか？

考察：会社 A の請求金額を取得する必要がある。

行動：GetInvoice[A]

結果：2000

考察：会社 B の請求金額を取得する必要がある。

行動：GetInvoice[B]

結果：1500

考察：会社 A の請求金額と会社 B の請求金額との間の差額を計算する必要がある。

行動：Diff[2000 1500]

結果：500

考察：よって、答えは 500。

行動：Finish[500]

質問：会社 B, C, D の請求金額の合計はいくらか？

考察：会社 B の請求金額を取得する必要がある。

行動：GetInvoice[B]

結果：1500

考察：会社 C の請求金額を取得する必要がある。

行動：GetInvoice[C]

結果：20000

考察：会社 D の請求金額を取得する必要がある。

行動：GetInvoice[D]

結果：6700

考察：会社 B, C, D の請求金額の合計を計算する必要がある。

行動：Total[1500 20000 6700]

結果：28200

考察：よって、答えは 28200。

行動：Finish[28200]

質問：会社 C の請求金額と会社 A, D の請求金額の合計との間の差額はいくらか？

考察：会社 C の請求金額を取得する必要がある。

行動：GetInvoice[C]

結果：20000

考察：会社 A の請求金額を取得する必要がある。

行動：GetInvoice[A]

結果：2000

考察：会社 D の請求金額を取得する必要がある。

行動：GetInvoice[D]

結果：6700

考察：会社 A, D の請求金額の合計を計算する必要がある。

行動：Total[2000 6700]

結果：8700

考察：会社 C の請求金額と会社 A, D の請求金額の合計との間の差額を計算する必要がある。

行動：Total[20000 8700]

結果：11300

考察：よって、答えは 11300。

行動：Finish[11300]

質問：会社 C, F の請求金額の合計と会社 A, E の請求金額の合計との間の差額はいくらか？

やっていること(1)

質問：会社 A の請求金額と会社 B の請求金額との間の差額はいくらか？

考察：会社 A の請求金額を取得する必要がある。

行動：GetInvoice[A]

結果：2000

考察：会社 B の請求金額を取得する必要がある。

行動：GetInvoice[B]

結果：1500

考察：会社 A の請求金額と会社 B の請求金額との間の差額を計算する必要がある。

行動：Diff[2000 1500]

結果：500

考察：よって、答えは 500。

行動：Finish[500]

質問：会社 B, C, D の請求金額の合計はいくらか？

考察：会社 B の請求金額を取得する必要がある。

行動：GetInvoice[B]

結果：1500

考察：会社 C の請求金額を取得する必要がある。

行動：GetInvoice[C]

結果：20000

考察：会社 D の請求金額を取得する必要がある。

行動：GetInvoice[D]

結果：6700

考察：会社 B, C, D の請求金額の合計を計算する必要がある。

行動：Total[1500 20000 6700]

結果：28200

考察：よって、答えは 28200。

行動：Finish[28200]

質問：会社 C の請求金額と会社 A, D の請求金額の合計との間の差額はいくらか？

考察：会社 C の請求金額を取得する必要がある。

行動：GetInvoice[C]

結果：20000

考察：会社 A の請求金額を取得する必要がある。

行動：GetInvoice[A]

結果：2000

考察：会社 D の請求金額を取得する必要がある。

行動：GetInvoice[D]

結果：6700

考察：会社 A, D の請求金額の合計を計算する必要がある。

行動：Total[2000 6700]

結果：8700

考察：会社 C の請求金額と会社 A, D の請求金額の合計との間の差額を計算する必要がある。

行動：Total[20000 8700]

結果：11300

考察：よって、答えは 11300。

行動：Finish[11300]

質問：会社 C, F の請求金額の合計と会社 A, E の請求金額の合計との間の差額はいくらか？

やっていること(2)

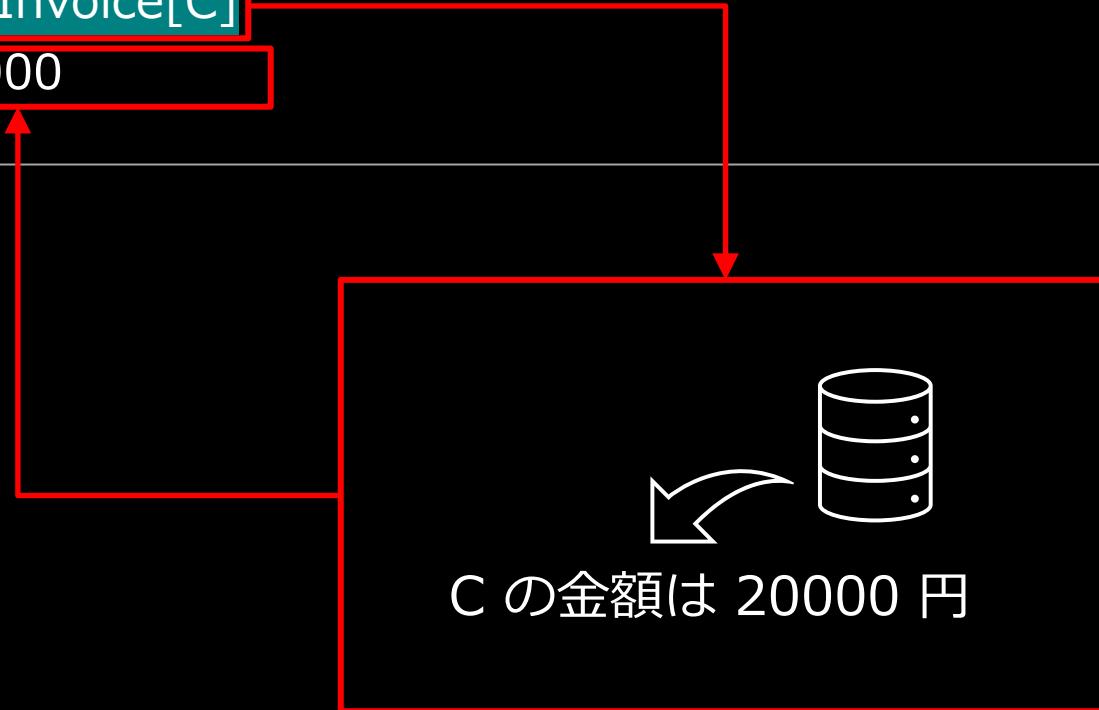
... few-shot prompt (略)

質問：会社 C, F の請求金額の合計と会社 A, E の請求金額の合計との間の差額はいくらか？

考察：会社 C の請求金額を取得する必要がある。

行動 : GetInvoice[C]

結果 : 20000



やっていること(3)

... *few-shot prompt* (略)

質問：会社 C, F の請求金額の合計と会社 A, E の請求金額の合計との間の差額はいくらか？

考察：会社 C の請求金額を取得する必要がある。

行動：GetInvoice[C]

結果：20000

考察：

やっていること(3)

... few-shot prompt (略)

質問：会社 C, F の請求金額の合計と会社 A, E の請求金額の合計との間の差額はいくらか？

考察：会社 C の請求金額を取得する必要がある。

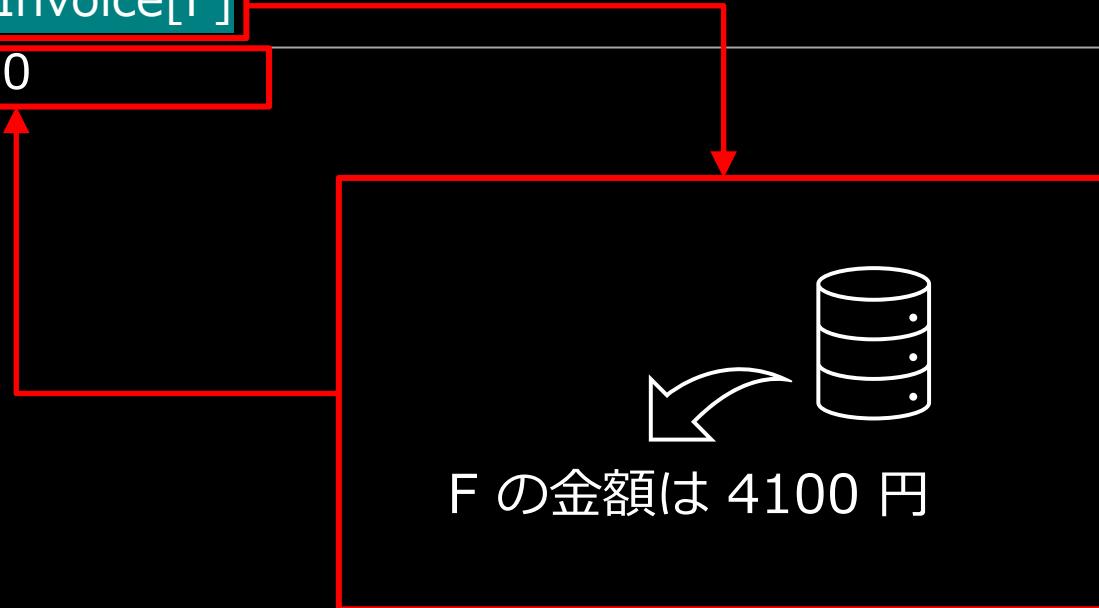
行動：GetInvoice[C]

結果：20000

考察：会社 F の請求金額を取得する必要がある。

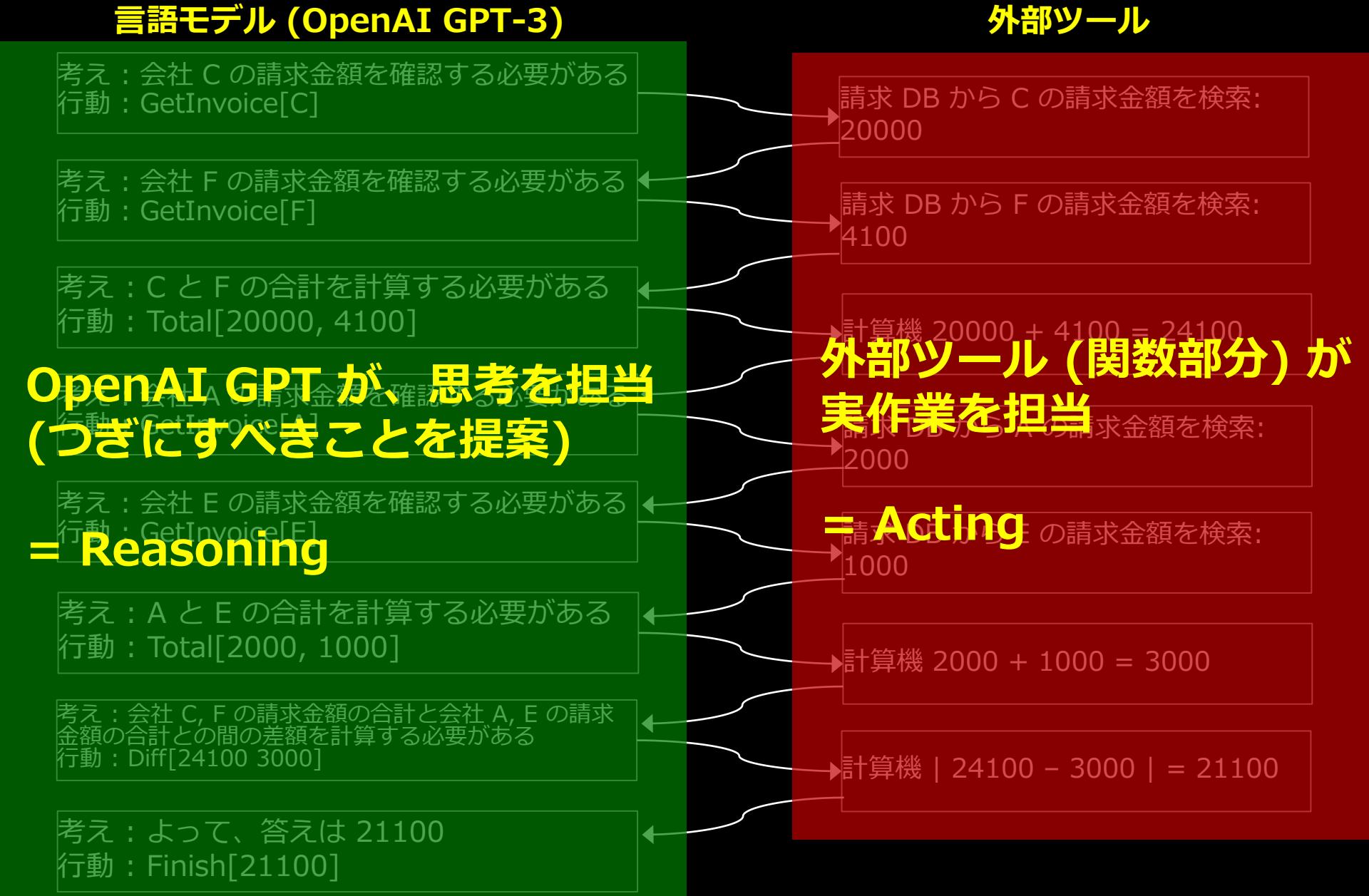
行動：GetInvoice[F]

結果：4100



以降、答えに到達するまで、外部ツールとの連携を繰り返す ...

やっていること (全体)



DEMO (PAL)

LangChain がサポートすること (抜粋)

QA, Summarization 等の一般タスク

Conversation Memory

Stream 開発 (および Async)

Knowledge Graph

Embedding, および各種 DB との連携

ReAct / MRKL 実装

開発

(FAISS, Chroma, Redis など各種サポート)

RDB, Wikipedia, Serp API 等との連携
ツール

HyDE (Hypothetical Document
Embeddings)

Moderation (Abuse 検出)

PAL による構築

など

要約 / 質問応答におけるチャunk分割

プロンプト エンジニアリング ガイド

1. 明確に指示 (instruction) を与える



2. 複雑なタスクは、より簡潔なサブタスクに分割



3. 指示 (instruction) をタスクに見合うように構造化する



4. 回答 (answering) の前に、プロンプトを使ってモデルに説明する



5. 可能な答え (複数) の正しさを確認して、それらを合わせる



6. 複数のアウトプットを生成して、モデルに一番良いものを選ばせる



7. パフォーマンス最大化のためカスタムモデルの Fine-Tuning をおこなう



GPT-3モデルを自分のタスクに合わせる方法

モデルの勾配更新不要

Zero-Shot

タスクの自然言語による記述だけで答えを予測する

One-Shot

タスクの説明に加え、モデルにはタスクの一例を与える

Few-Shot

タスクの説明に加えて、モデルはタスクの複数の例を与える

Fine Tuning

このモデルは、大規模なタスク例を用いた勾配更新の繰り返しにより学習される。

トレーニングデータの作成とアップロード



微調整された新しいモデルをトレーニングする



微調整されたモデルを使う

1.

プロンプトデザインよりも高品質な仕上がり

2.

プロンプトに収まりきらないほど多くの例題をトレーニングする能力

3.

プロンプトの短縮によるトークンの節約

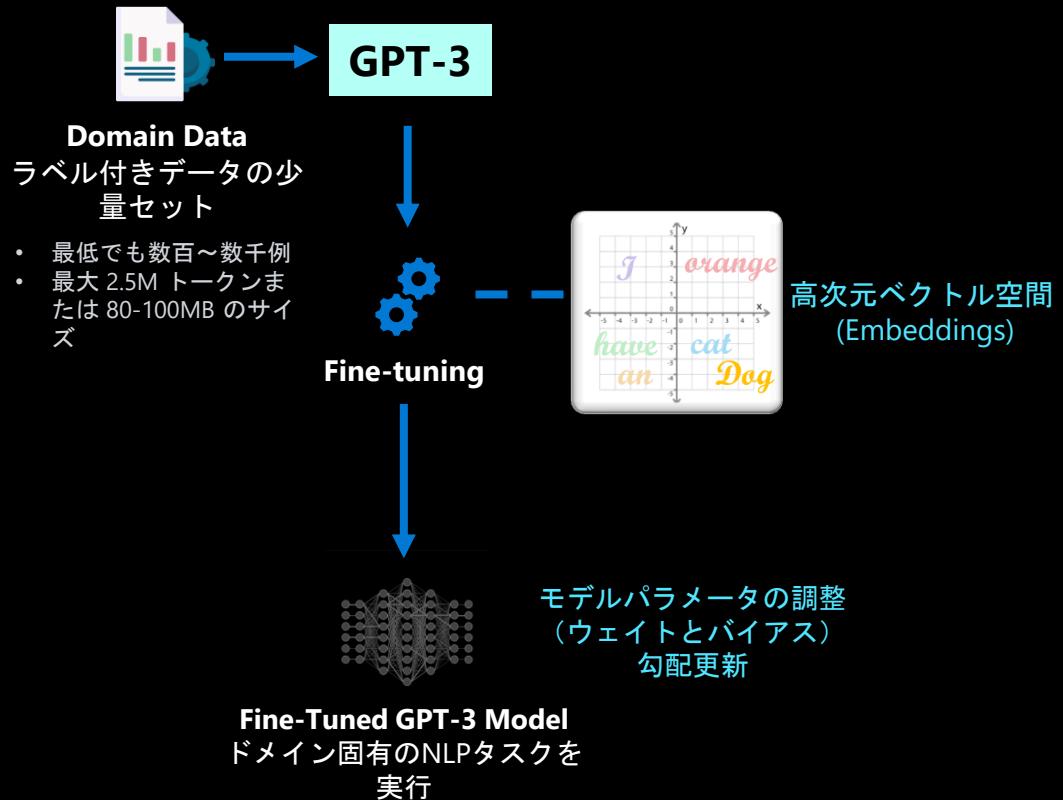
4.

低レイテンシーエクエスト

Fine-tuning

特定ドメインデータによるモデル適応

Fine-Tuning



Fine-tuning の結果、重みとバイアスが更新さ
れた新しいモデルが生成されます。

これは、モデルの重みとバイアスが更新されな
い Few-shot learning とは対照的である。



Fine-tuning が必要な時

モデルが嘘の発言（幻覚）をしている場合、幻覚の緩和のために使用
モデルの結果の精度が顧客の要求を満たさない

ファインチューニングにより、API を通じて利用可能なモデルをより多く提供することができます：

- ・ プロンプトデザインよりも高品質な結果
- ・ プロンプトに収まりきらないほどの例題をトレーニングする能力
- ・ 低レイテンシーエンド

ファインチューニングは、プロンプトに収まらないほど多くの例で学習することで、Few-shot learning よりも改善され、様々なタスクでより良い結果を得ることができます。



Fine-Tuning で準備するデータセット

ファインチューニングデータセットは JSONL 形式であること
(ただし、CSV, TSV, XLSX, JSON からの変換可能)

1つの入力 ("プロンプト") とそれに関連する出力 ("完了") からなる訓練例のセット。

- ・ 例えば、分類タスクの場合、プロンプトは問題文、完了は対象クラス
- ・ テキスト生成タスクの場合、プロンプトは指示/質問/要求、完了はテキストの実際のデータ(Ground truth)

```
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}  
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}  
{"prompt": "<prompt text>", "completion": "<ideal generated text>"}
```

Fine-tuning の実行

UI (OpenAI Studio), REST, Python SDK から実行が可能
ジョブが作成される (ジョブは途中でキャンセル可能)

The image displays four sequential screenshots of the 'Create customized model' wizard in the Azure OpenAI Studio:

- Step 1: Base model**

Shows the selection of a base model (ada, babbage, curie, davinci, code-cushman-001) and the choice to 'Review and train'.
- Step 2: Validation data**

Shows the selection of validation data (Local file or Azure blob). A file upload dialog is shown, prompting for a JSONL file.
- Step 3: Training data**

Shows the selection of training data (Local file or Azure blob). A file upload dialog is shown, prompting for a JSONL file.
- Step 4: Advanced options**

Shows various advanced parameters:
 - Default vs Advanced (Advanced selected)
 - Number of epochs: 2
 - Batch size: 4
 - Learning rate multiplier: 1
 - Prompt loss weight: 0.1

Fine-Tuning に関するハイパーパラメータ

パラメータ	内容	推奨
<code>n_epochs</code>	モデル学習のエポック数。 1 エポックで学習データを 1 サイクル通して学習する	4 から開始し、データセットが小さい場合にはより多くのエポックを設定し、大きなデータセットでは小さなエポック数を設定する。 学習精度が良くない場合 (underfitting) は、この値を増やして試してみる。学習精度は良いが、Validation の精度が低い場合、過学習となっている可能性があるため、この値を低くして試す。
<code>batch_size</code>	学習時のバッチサイズ。バッチサイズは、学習時の 1 回の forward / backward 処理で使う学習のサンプル数。	学習用のデータサイズの 0.01% から 4% 程度で良いパフォーマンスが期待できる。一般に、データセットが多い場合には、バッチサイズも大きくすると良いとされている。
<code>learning_rate_multiplier</code>	この Fine-tuning の Learning Rate Multiplier は、事前学習で使用された Learning Rate の Multiplier (倍数) を意味している。	良い結果を出すために、この値は 0.02 から 0.2 の範囲で実験するのが推奨。 経験則として、バッチサイズが大きい場合には、大きめの Learning Rate を設定すると良い。 また、経験則として、この <code>learning_rate_multiplier</code> は、前述のエポック数やバッチサイズよりも学習結果に与えるインパクトは小さい。
<code>prompt_loss_weight</code>	prompt token の loss の割合。この値により、モデルが prompt 生成のために学習する割合 (completion token の学習を 1.0 とした場合の割合) を制御できる。completion token が短いときに、この値を増やすことで、prompt と completion の学習をバランスさせることが可能。	Fine-tuning では、内部で、prompt と completion の双方のテキスト生成の学習をおこなう。 prompt テキストと completion テキストの学習の仕方の違いは、各 completion token と比較して各 prompt token のほうが、より少なく評価される点である。この学習の割合を制御する方法として、 <code>prompt_loss_weight</code> が使用できる。(この <code>prompt_loss_weight</code> の既定値は 0.1) 例えば、もし prompt が completion と比べて非常に長い場合、 <code>prompt_loss_weight</code> を減らすことで、prompt 学習に対する過剰な評価を避けることができる。 経験則として、この <code>prompt_loss_weight</code> は、前述のエポック数やバッチサイズよりも学習結果に与えるインパクトは小さい。



Fine-Tuning のベストプラクティス

Fine-tuning は、より質の高い例 (学習データ) でより良いパフォーマンスを発揮する。

プロンプトを使用するよりも優れたパフォーマンスを発揮するモデルをFine-tuning するには、少なくとも数百の高品質の例を提供する必要がある。

そこから、例の数が 2 倍になるごとに、性能は直線的に向上する傾向がある。例の数を増やすことは、通常、精度を向上させるための最も確実で最適な方法。

ファインチューニングのデータセットでは、間違ったラベルを完全に削除する。可能であれば不快な内容や不正確な内容を手動で確認し、データセットが大きい場合はできるだけ多くのランダムサンプルを確認する。

データサイズの微調整：高度なモデル (DaVinci) は、限られたデータ量でより良いパフォーマンスを発揮し、十分なデータ量があれば、すべてのモデルがうまくいく。※

※ ただし、現在、新規利用者は DaVinci の選択は不可

Fine-Tuning のためのガイドと例

Getting started with Fine-Tuning:

[How to customize a model with Azure OpenAI - Azure OpenAI | Microsoft Learn](#)

Fine-tuning—OpenAI API

<https://platform.openai.com/docs/guides/fine-tuning/general-best-practices>

Detailed examples of fine-tuning

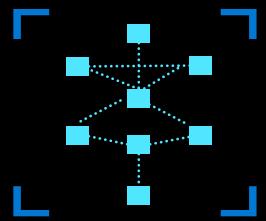
[\[PUBLIC\] Best practices for fine-tuning GPT-3 to classify text—Google Docs](#)

Using fine-tuning to improve truthfulness

[Fine-tuning a Classifier to Improve Truthfulness | OpenAI Help Center](#)

Embeddings Model

Embeddings



Embeddings とは、機械学習モデルやアルゴリズムが容易に利用できる特殊なデータ表現形式

Embeddings は、テキスト片の意味論的な意味（意味や意図、およびその背後にある知識や文脈）を情報密度で表現したもの

各 Embeddings は浮動小数点のベクトルであり、ベクトル空間における 2 つの Embeddings 間の距離は、元のフォーマットにおける 2 つの入力間の意味的類似性と相關するようになっている

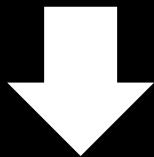
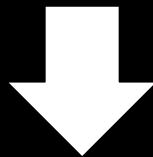
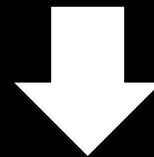
例えば、2 つのテキストが似ている場合、それらのベクトル表現も似ている

*Embeddings*により、コンテンツを "意味空間" にマッピングすること
が可能になる

中性子星は、大質量の
超巨大星が崩壊したコ
アである

星は、熱核融合によっ
て活動期のほとんどで
輝いています。

ブラックホールの存在
は、他の物質との相互
作用によって推察する
ことができる

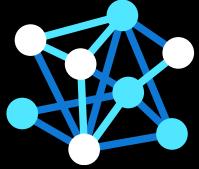


[15 34 24 13 ...]

[16 22 89 26 ...]

[20 13 31 89 ...]

Embedding models



異なる Azure OpenAI の Embeddings モデルは、特定のタスクが得意なように特別に作られています。

- **Similarity embeddings** は、2つ以上のテキスト間の意味的な類似性を捕らえるのに適しています。
- **Text search embeddings** は、短いクエリに対して長い文書が関連しているかを測定するのに役立ちます。
- **Code search embeddings** は、コードスニペットを埋め込んだり、自然言語検索クエリを埋め込むのに便利です。

text-embedding-ada-002 (V2) 推奨

テキスト検索

モデル	パフォーマンス
text-embedding-ada-002	53.3
text-search-davinci-* -001	52.8
text-search-curie-* -001	50.9
text-search-babbage-* -001	50.4
text-search-ada-* -001	49.0

データセット: BEIR (ArguAna, ClimateFEVER, DBpedia, FEVER, FiQA2018, HotpotQA, NFCorpus, QuoraRetrieval, SciFact, TRECCovid, Touche2020)

Model Gen.	Tokenizer	Max Tokens	Dimension	Knowledge Cutoff
2	cl100k_base	8191	1536	Sep 2021

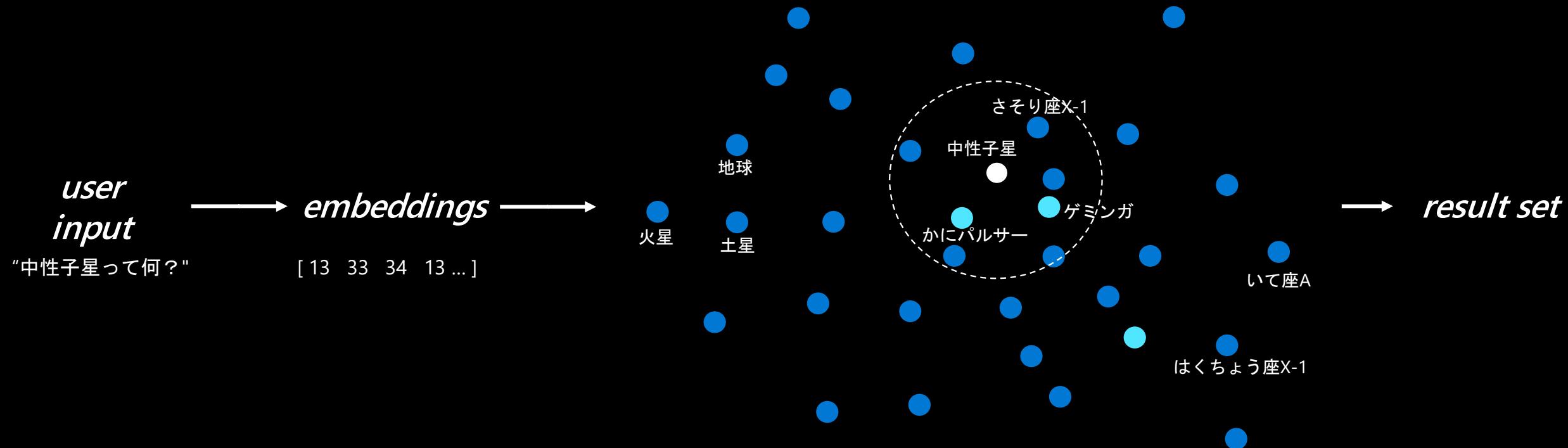
文の類似性

モデル	パフォーマンス
text-embedding-ada-002	81.5
text-similarity-davinci-001	80.3
text-similarity-curie-001	80.1
text-similarity-babbage-001	80.1
text-similarity-ada-001	79.8

データセット: SentEval (STS 2012–2016)

Embeddings を用いた類似性検索

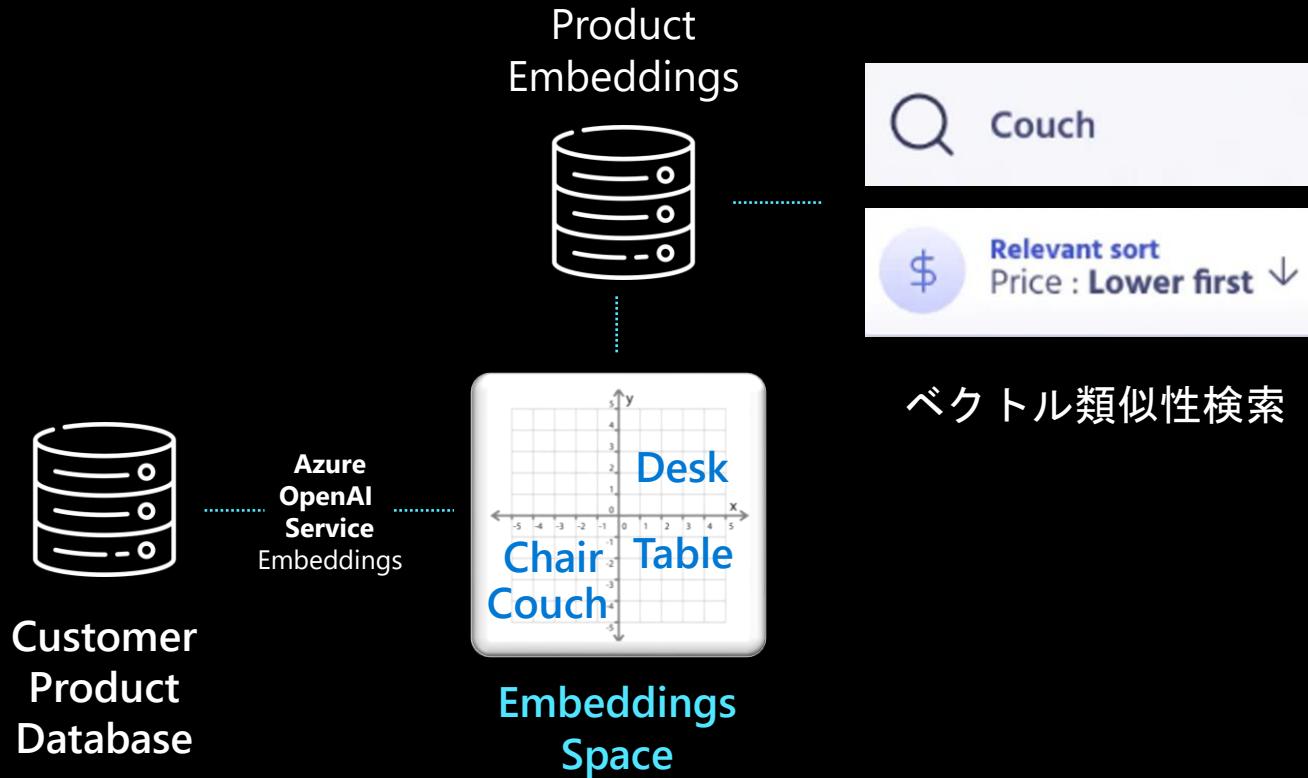
コンテンツを Embeddings としてエンコードすれば、あとはユーザー入力から Embeddings を取得し、それを使って最も意味的に類似したコンテンツを探し出すことができます。





リテール業界での利用例

ベクトル類似性検索



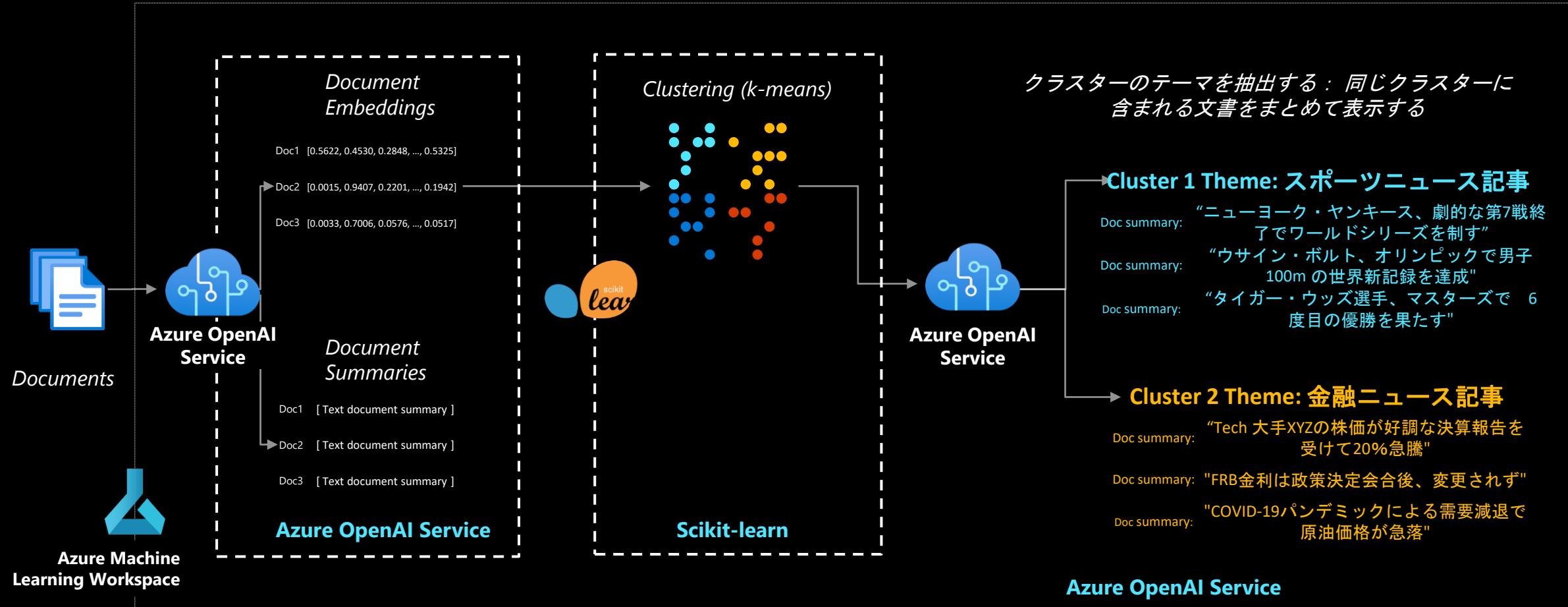
ベクトル類似性検索





Document Clustering, and Cluster Theme Generation

文書の類似性埋め込みを要約・抽出、類似クラスターを探索、クラスター・テーマを生成



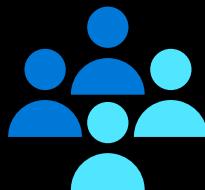
Codex Model



APIへのアクセス性を高める



ソフトウェア開発を加速



Coding for Everyone

基本モデルから派生し、自然言語とコードの両方でトレーニング(数十億行のコード)

複数のプログラミング言語をサポート

Python, C#, SQL, Java, JavaScript, TypeScript, Go, Perl, PHP, Ruby, Swift, Shell (bash)

Codex



JS fetch_pic.js

```
1 const fetchNASAPi
2   return fetch('
3     method: 'GET'
4     headers: {
5       'Content-Type': 'application/json'
6     },
7   })
8   .then(response =>
9     .then(json =>
10       return json
11     });
12 }
```

Copilot

複数のタスク:

コメント→コード

オートコンプリート機能または次の行を提案(コンテキスト内)

ナレッジ検索 (API またはライブラリの呼び出し)

コードのドキュメント化 (コメント)

リファクタリング

ユースケース

自然言語からコードへ

コードのドキュメント化 (コメント)

リファクタリング

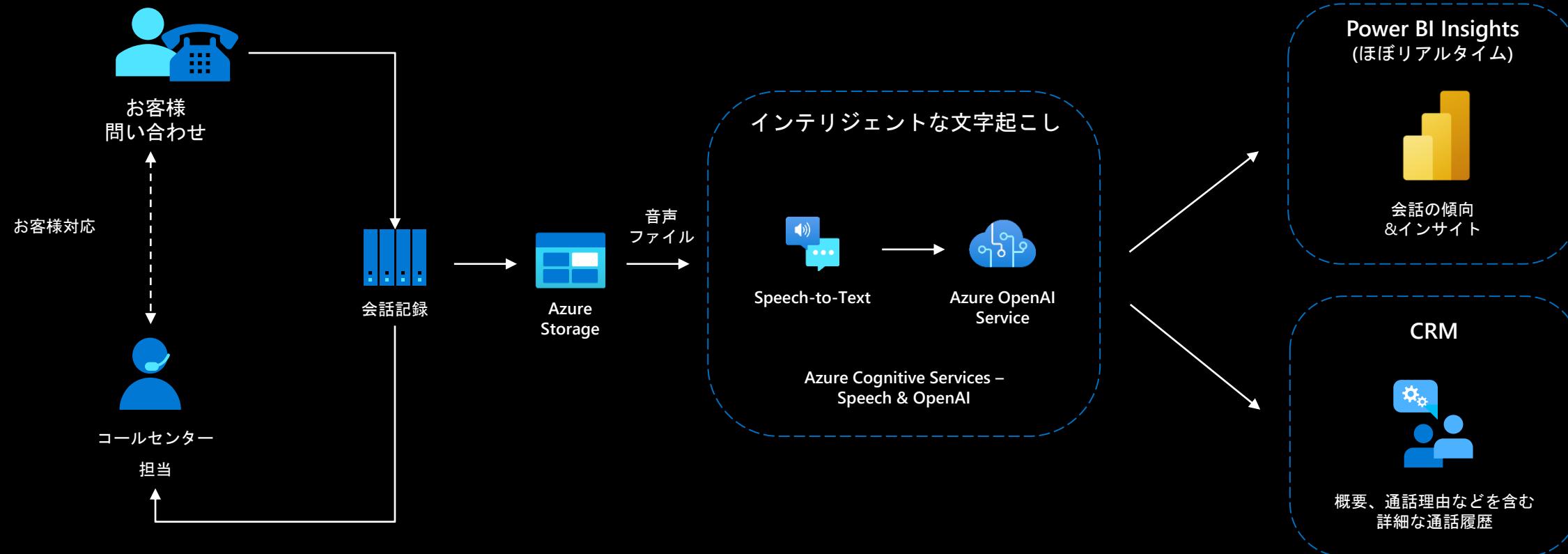
コードを自然言語へ

自然言語から SQL へ

アーキテクチャ

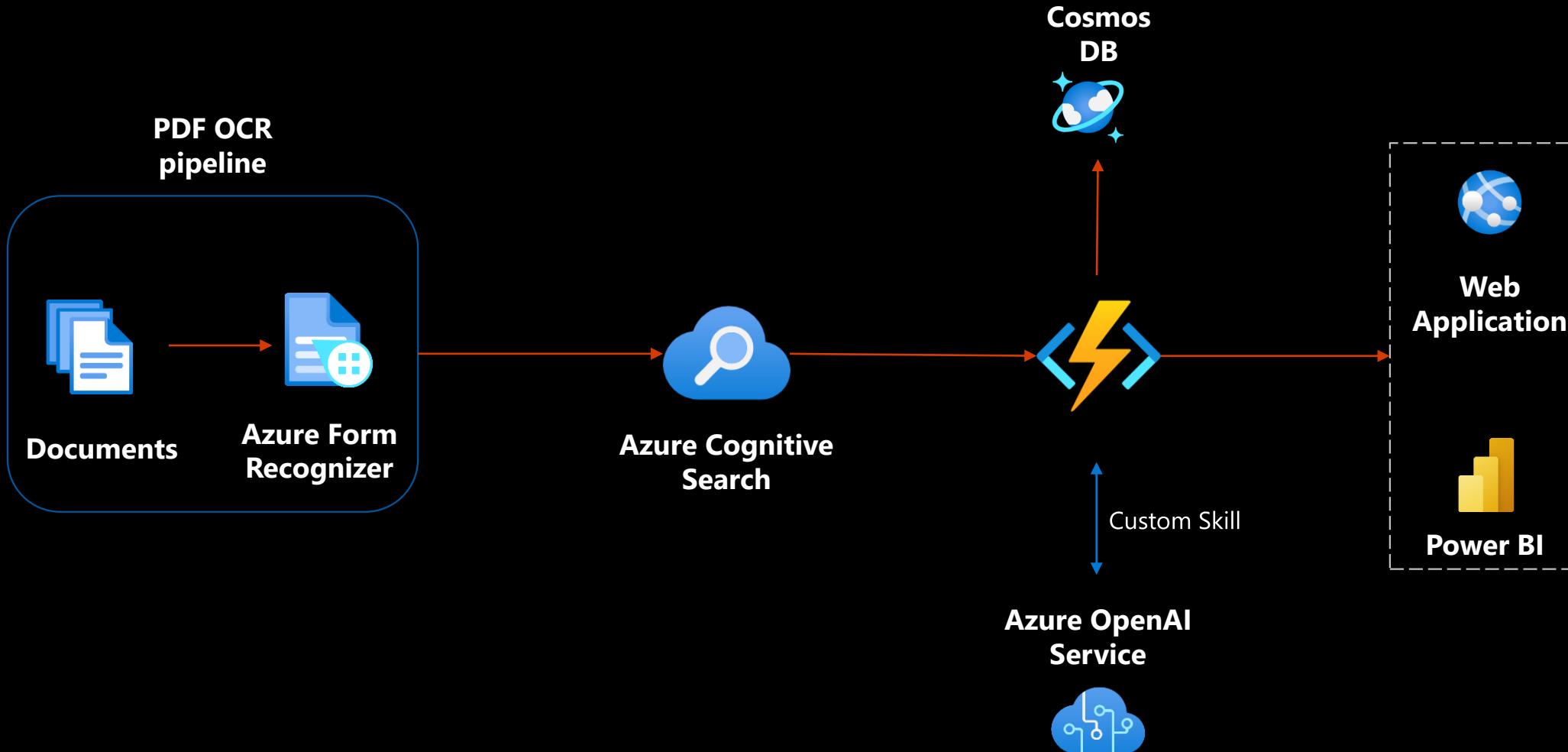
Speech API と OpenAI を使用したコールセンター会話記録分析

通話記録から豊富な分析情報を抽出する



Document Process Automation

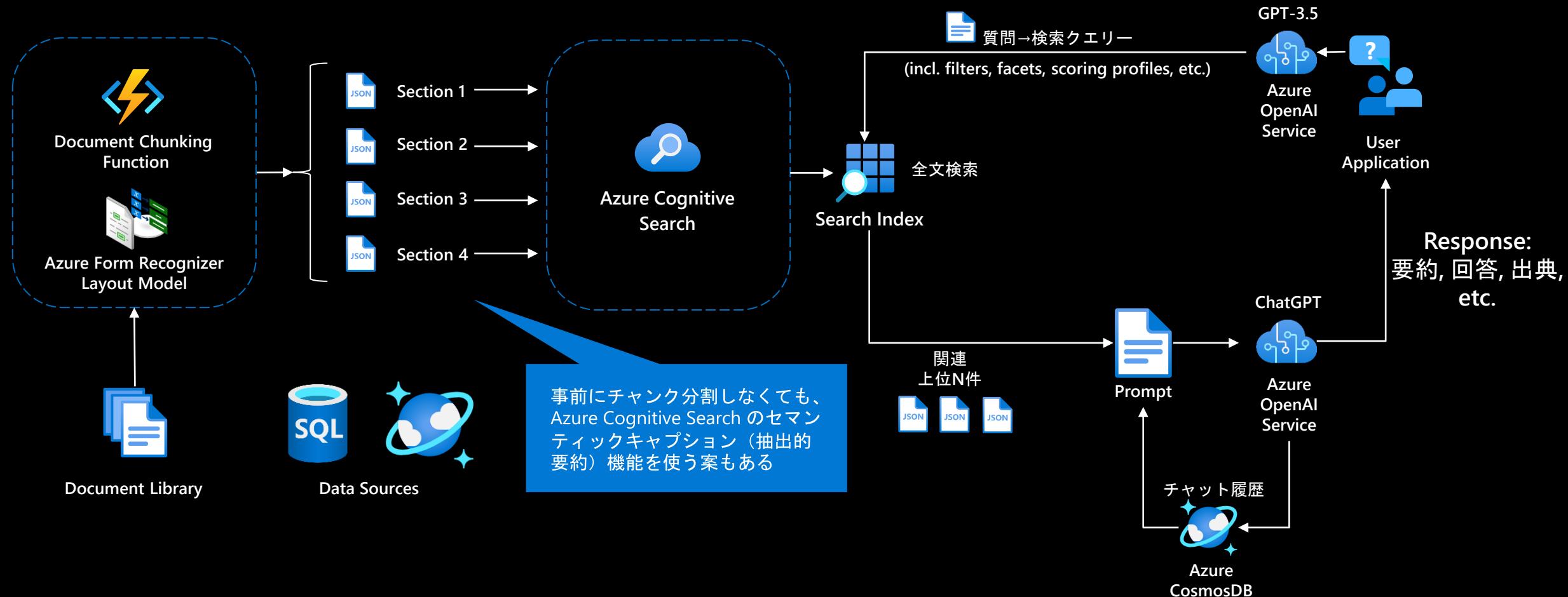
ドキュメントから豊かなインサイトを抽出し、要約する



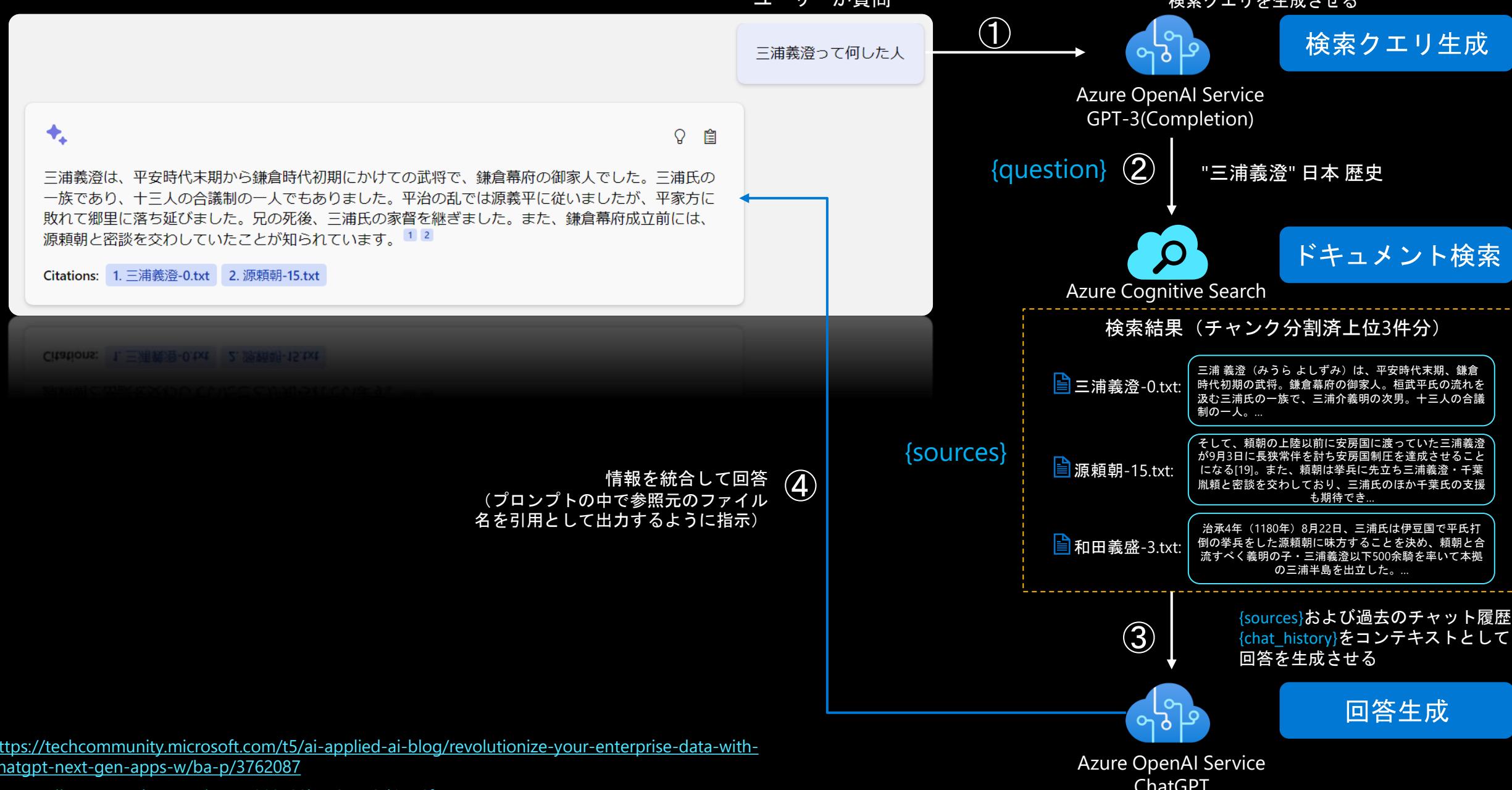


AI-Powered Q&A over Enterprise Data Sources

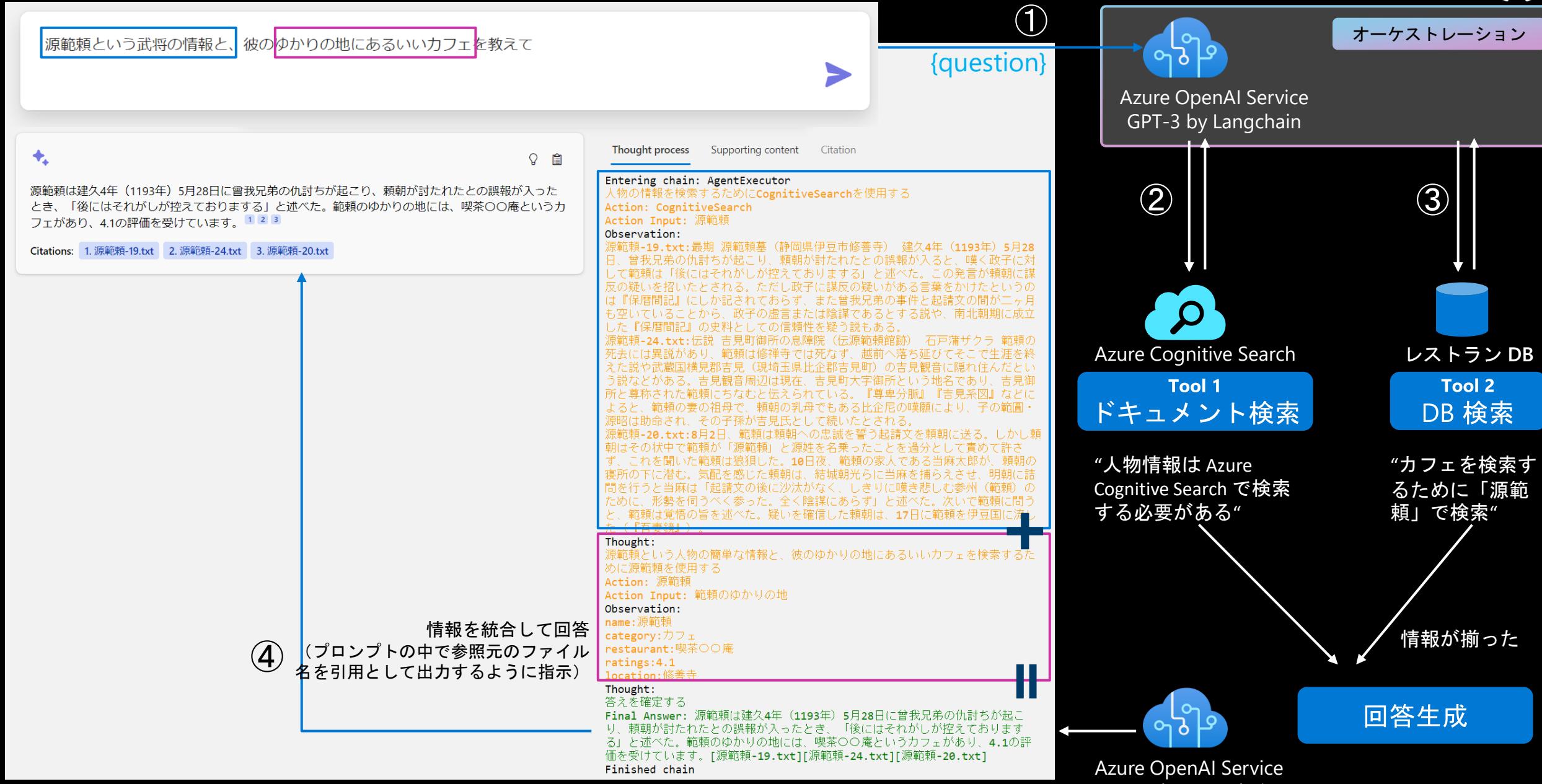
企業独自のデータを使って Q&A チャットボットを構築



Chat Read-Retrieve-Read Approach

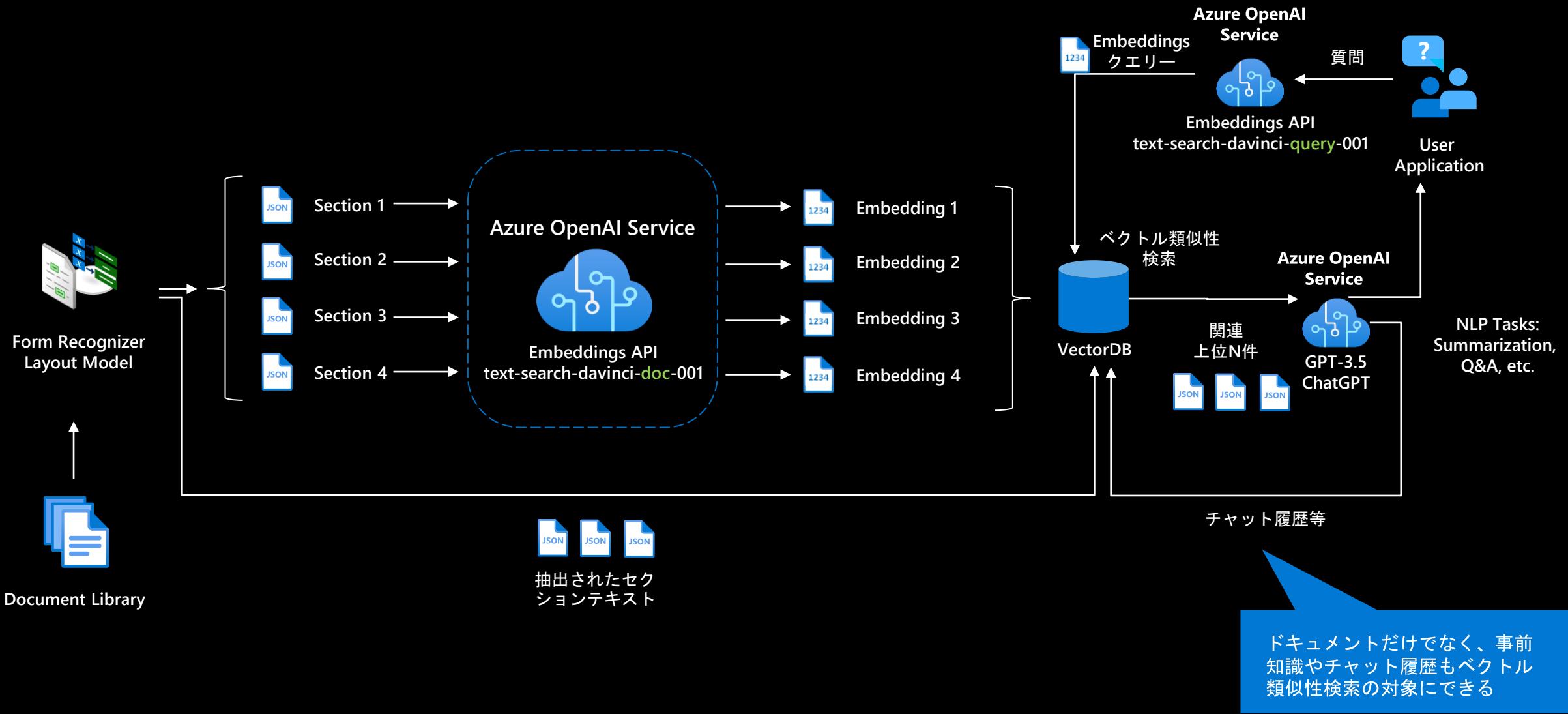


Modular Reasoning, Knowledge and Language(MRKL)



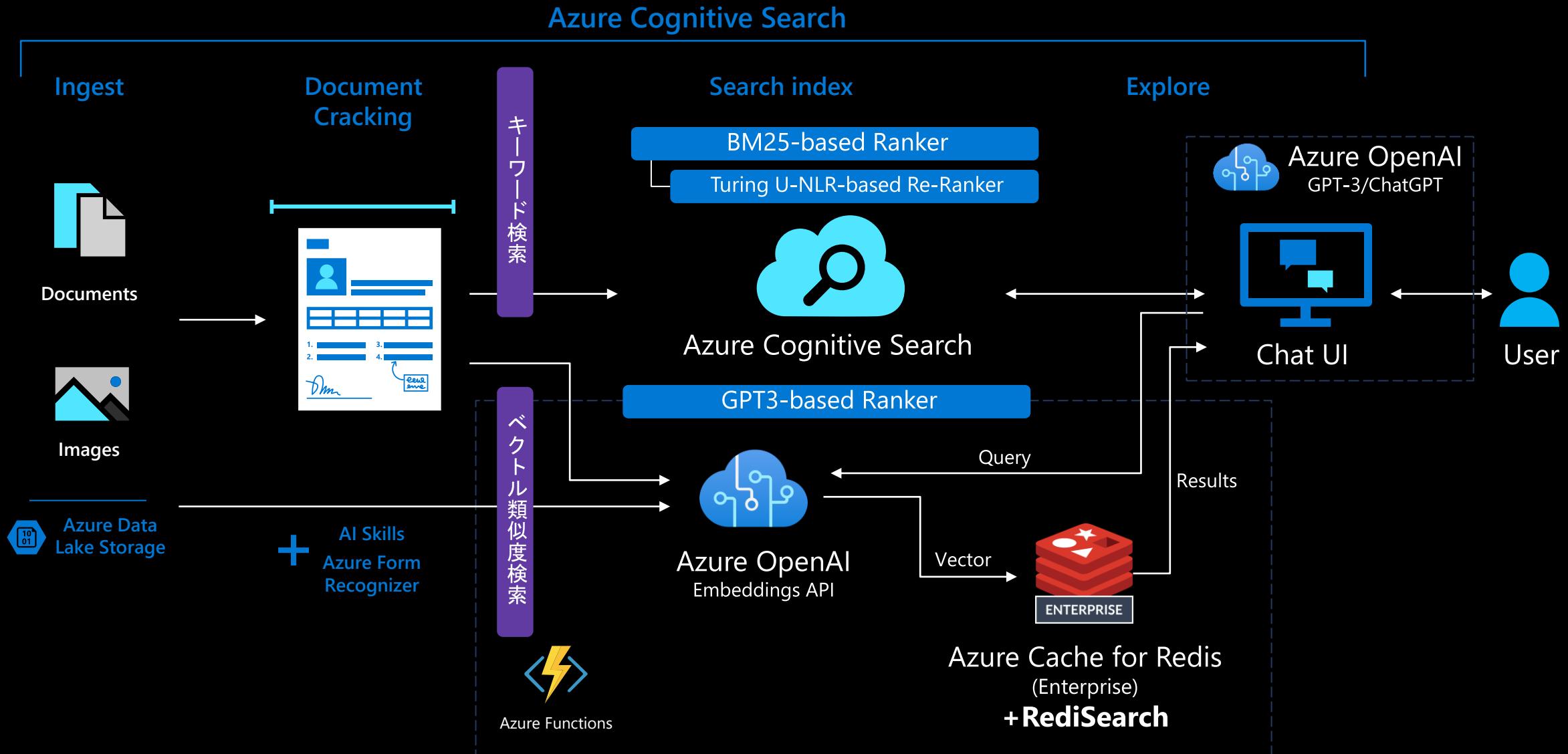


Q&A with Semantic Answering over Document Library





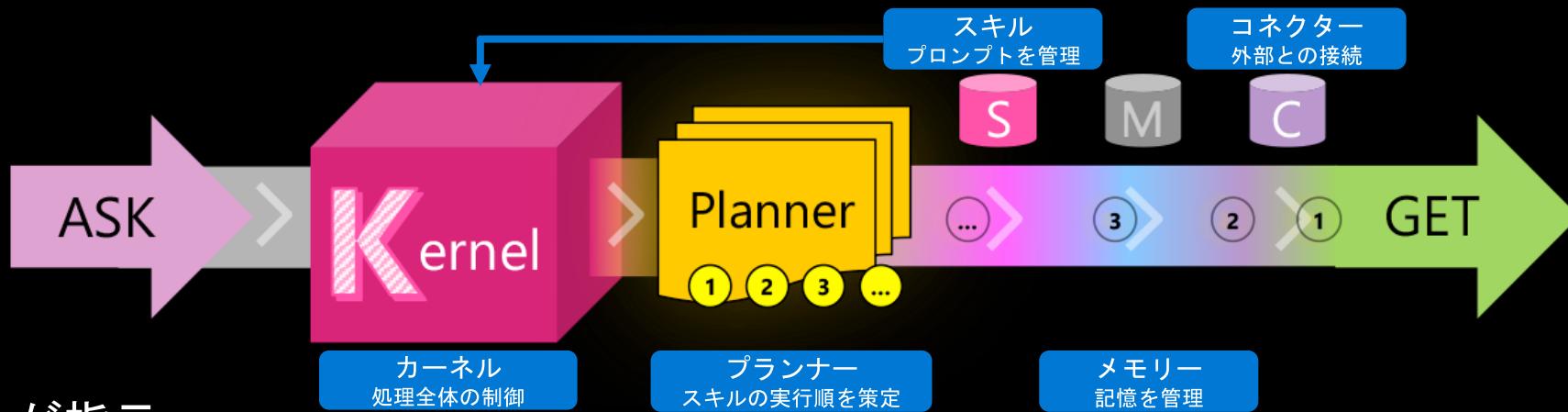
Keyword Search vs Vector Similarity Search on Azure



Semantic Kernel(SK)

<https://github.com/microsoft/semantic-kernel>

Semantic Kernel(SK)



ユーザーが指示

明日はバレンタインデーです。デートのアイデアをいくつか考えなければいけません。彼女はシェイクスピアが好きなので、彼のスタイルで書いてください。あとそのアイデアを私の大切な人にメールして。

実行計画(GPT-3)

```

<plan>
  START
    <function.WriterSkill.Brainstorm
      input="バレンタインデーのデートアイデア" setContextVariable="IDEAS"/>
    <function.ShakespeareSkill.shakespeare
      input="$IDEAS" setContextVariable="SHAKESPEARE_IDEAS"/>
    <function.WriterSkill.EmailTo
      input="私の大切な人" setContextVariable="$SHAKESPEARE_IDEAS"
      sender="Me" appendToResult="RESULT_EMAIL"/>
  GOAL
</plan>
  
```

スキル



実行



スキル

セマンティック関数

ネイティブ関数

skprompt.txt

```
sk_prompt = """"  
{{\$input}}  
  
上記の内容を3つのポイントに要約してください。  
"""
```

config.json

```
prompt_config = sk.PromptTemplateConfig.from_completion_parameters(  
    max_tokens=1000, temperature=0.2, top_p=0.5  
)
```

ファイルでもインラインでも可

登録

```
kernel.register_semantic_function("SummarySkill", "Summary",  
    function_config)
```

```
kernel.import_semantic_skill_from_directory(skills_directory, "Summary  
Skill")
```

text_skill.py

```
from semantic_kernel.skill_definition import sk_function  
  
class TextSkill:  
  
    @sk_function(description="Trim whitespace from the start and end of a string.")  
    def trim(self, text: str) -> str:  
        """  
        Trim whitespace from the start and end of a string.  
  
        Example:  
        SKContext["input"] = " hello world "  
        {{text.trim $input}} => "hello world"  
        """  
        return text.strip()
```

登録

```
kernel.import_skill("text", TextSkill())
```



プランナー

PlannerSkill Prompt

与えられた目標を満たすために、**ステップバイステップでXML**プランを作成する。
プランを作成するには、以下の手順で行います：

1. <goal>から一連の<functions>として<plan>を作成する。
2. 与えられた目標に必要な関数のみを使用する。
3. 関数は、'input'と'output'を持つ。
4. 各関数の'output'は、後続の<function>の'input'として自動的に渡される。
5. 'input'は、前の関数の'output'を消費する場合は、指定する必要はない。
6. ある<function>の'output'を保存して、後の<function>に渡すには、
`<function.{FunctionName} ... setContextVariable: "$<UNIQUE_VARIABLE_KEY>" />`とする。
7. <function>から'output'を保存し、プランの結果の一部として返すには、
`<function.{FunctionName} ... appendToResult: "RESULT_$<UNIQUE_RESULT_KEY>" />`とする。
8. ""if""と""else""タグのみを使用する。
9. ""if""と""else""タグは必ず閉じてください。
10. 比較演算子はリテラルでなければならない。
11. プラン末尾のXMLコメントに""END""を付加する。
12. [AVAILABLE FUNCTIONS]のみを使用する。

[AVAILABLE FUNCTIONS]

`[$available_functions]`

[END AVAILABLE FUNCTIONS]

<goal>`[$input]`</goal>

+Few-shot Samples...

明日はバレンタインデーです。デートのアイデアをいくつか考えなければいけません。彼女はシェイクスピアが好きなので、彼のスタイルで書いてください。あとそのアイデアを私の大切な人にメールして。



与えたスキルの説明等から、必要なスキルのみを選定する

Everything.Summarize:

description: summarize input text

inputs:

- input: the text to summarize

LanguageHelpers.TranslateTo:

description: translate the input to another language

inputs:

- input: the text to translate

- translate_to_language: the language to translate to

EmailConnector.EmailTo:

description: email the input text to a recipient

inputs:

- input: the text to email

- recipient: the recipient's email address. Multiple addresses may be included if separated by ','.

スキルの関数一覧

メモリーとコネクター

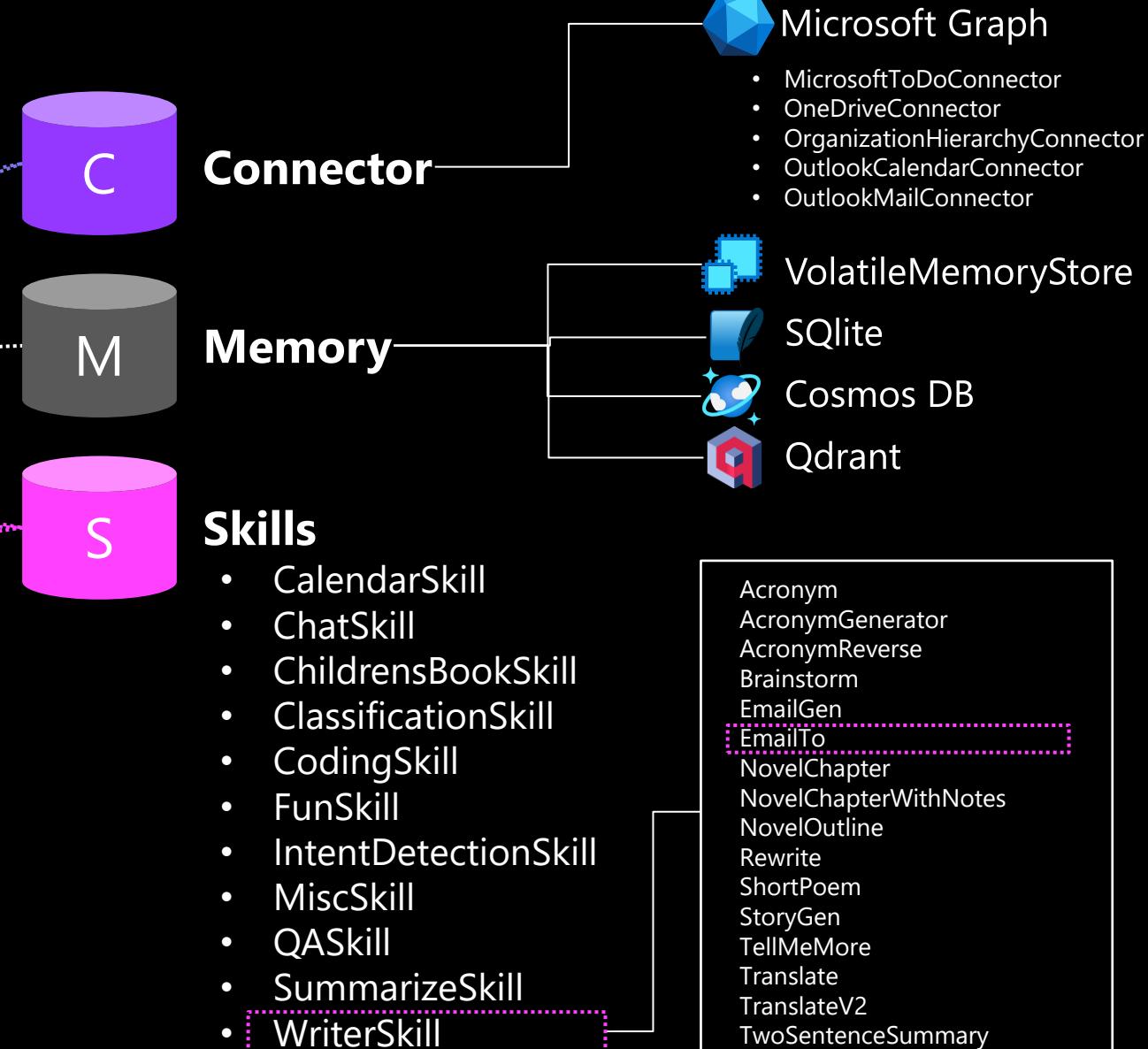
セマンティック関数

```
sk_prompt = """
My name: {{msgraph.GetMyName}}
My email: {{msgraph.GetMyEmailAddress}}
My hobbies: {{memory.recall "my hobbies"}}
Recipient: {{$recipient}}
Email to reply to:
===== {{${sourceEmail}}}
"""

以下を使ってメールの返信を作成します。
{{$input}}
```

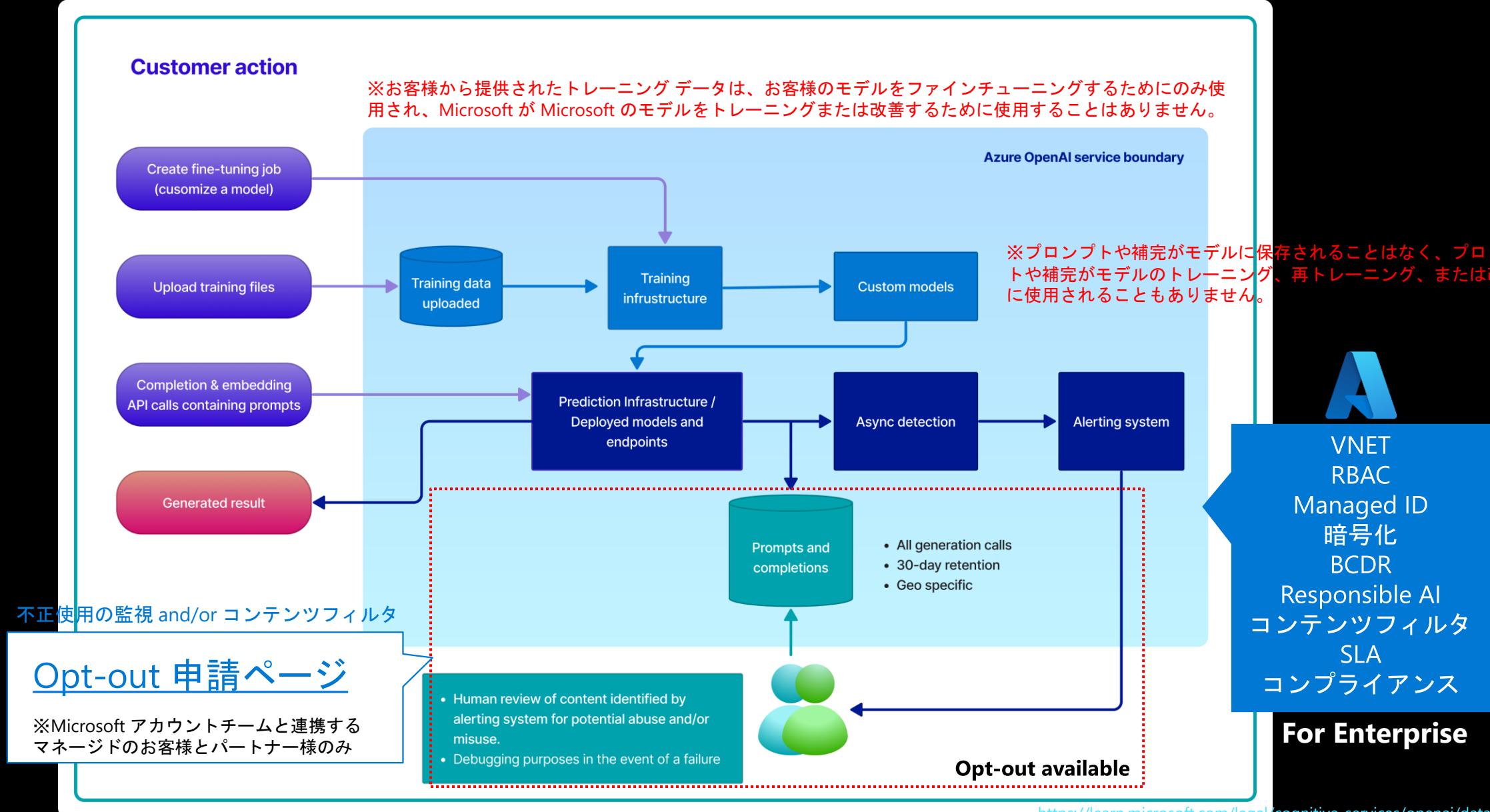
回答後に引用された元のメールを記載する。

....

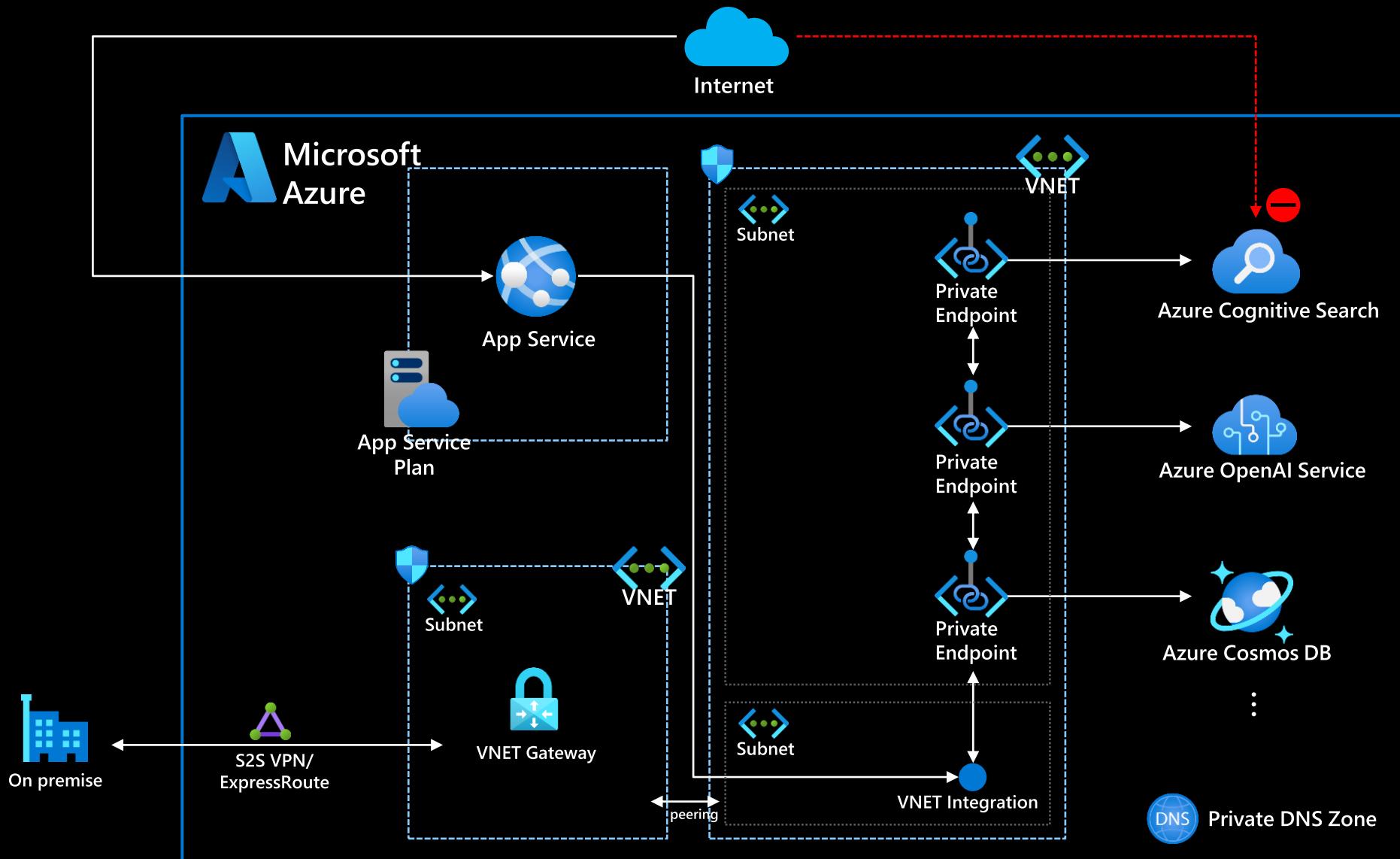


プライバシー、セキュリティ

Azure OpenAI データ、プライバシー、セキュリティ

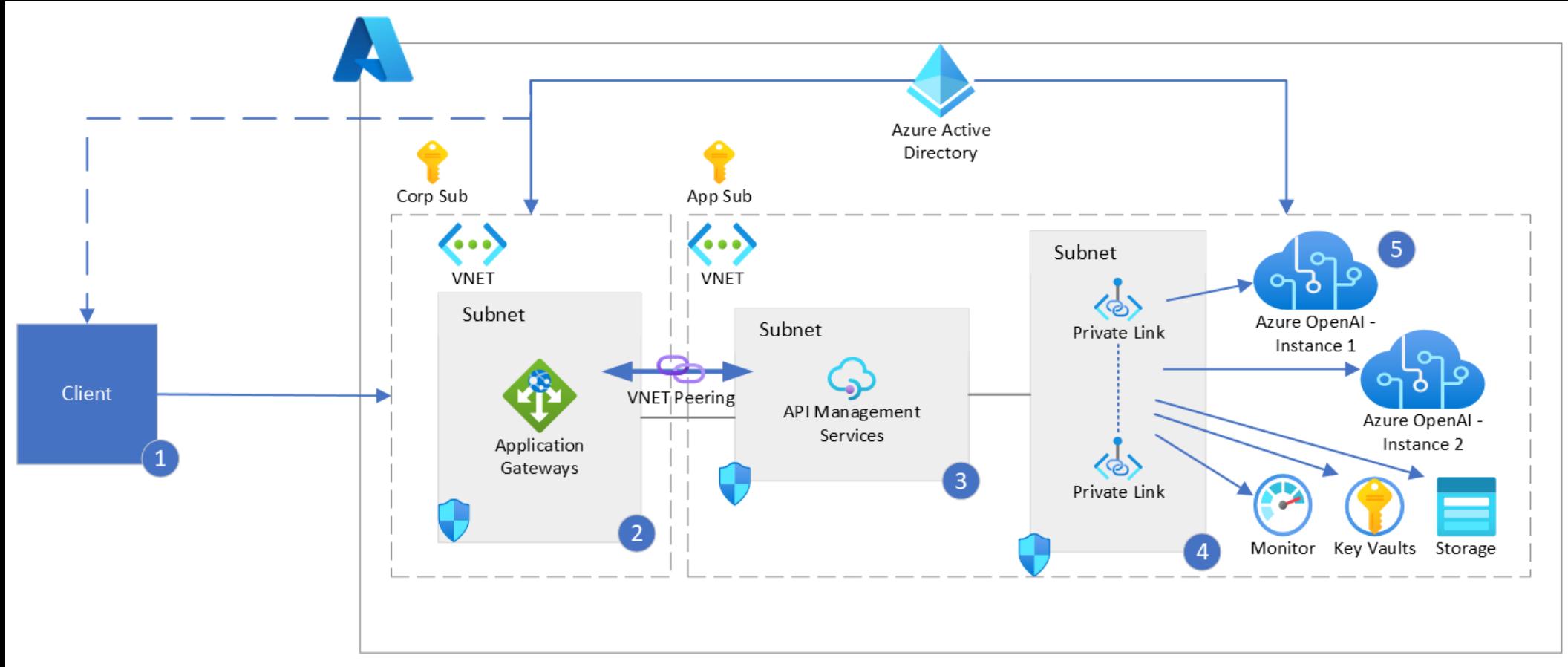


Azure OpenAI Service 閉域網構成





Azure OpenAI Service Enterprise Logging



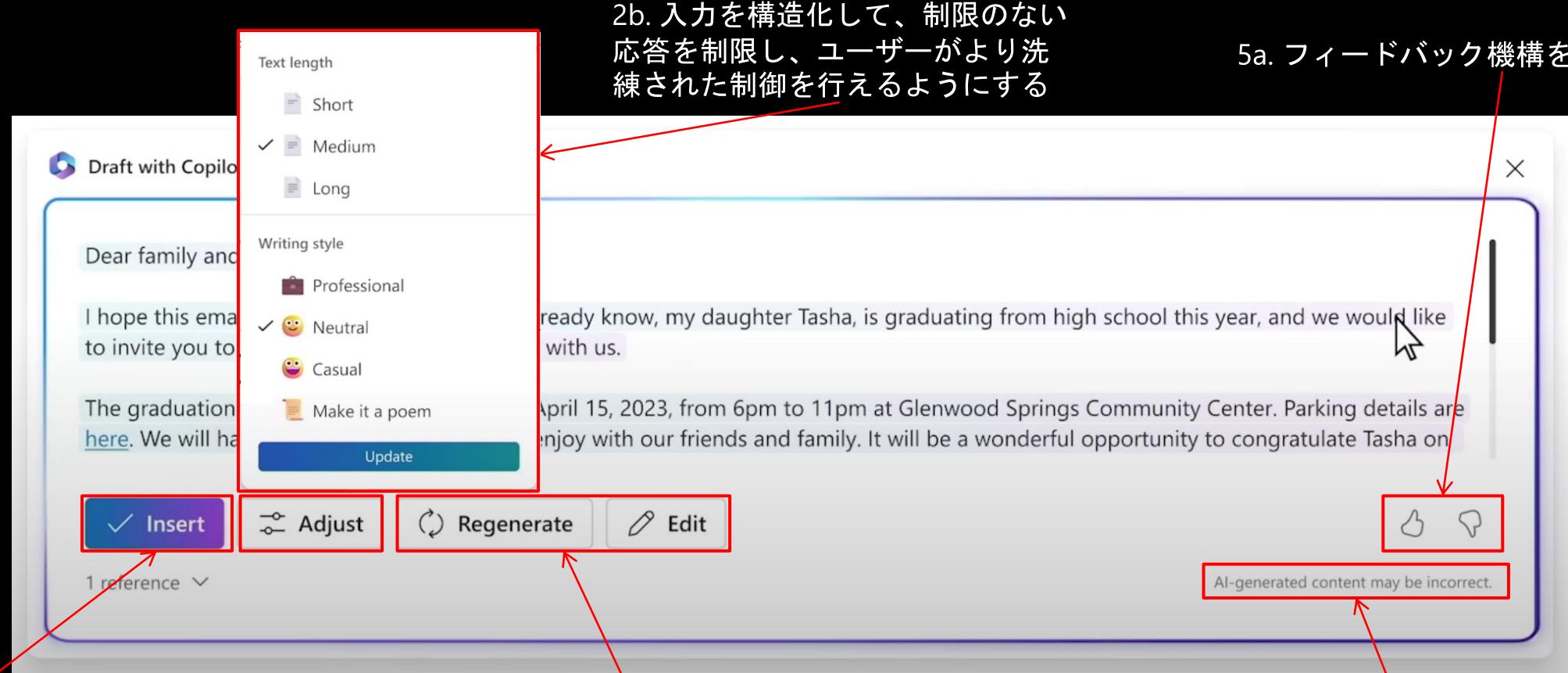
システム開発における考慮事項

モデルの幅広い知識とデータ、および動的コンテンツを生成する能力を考えると、アプリケーションでの責任ある使用を確保するために特別な注意を払う必要があります。

考慮事項	解説	緩和措置
制限のない、制約のないコンテンツ生成には適していません	ユーザーが任意のトピックに関するコンテンツを生成できるシナリオでは、攻撃的または有害なテキストが生成される可能性が高くなります。	プロンプト設計時に詳細に伝える アプリケーションのトピックを固定 モデルの品質を測定 入力と出力の長さ、構造、速度、およびソースを制限 人間による監視を確保 生成された出力を編集可能にする 生成された出力の潜在的な不正確さを強調 最終的な決定や内容についてユーザーに説明責任 自動化する方法を制限 生成されたコンテンツにおけるAIの役割を開示
最新で事実上正確な情報が重要であるシナリオには適していません（人間のレビュー者がいる場合や、モデルを使って独自の文書を検索し、シナリオへの適合性を検証している場合を除く）	モデルの学習日以降に発生した事象に関する情報を持っておらず、いくつかのトピックに関する知識が不足している可能性があり、常に事実に基づいて正確な情報を生成することは限りません。	入力と出力に技術的な制限を実装 入力と出力の長さを制限 入力を構造化して、制限のない応答を制限 検証済みの信頼できるソース資料からの出力を返します ブロックリストとコンテンツモデレートを実装 レート制限(API呼び出しの頻度と量)を設定
システムの使用または誤用が、個人的身体的または精神的な重大な傷害につながるようなシナリオを避ける	例えば、患者を診断したり、薬を処方したりするシナリオは、重大な被害をもたらす可能性があります。	ユーザーを認証 アプリケーションを徹底的にテスト フィードバック機能をUIに組み込む
システムの使用または誤用が、人生の機会や法的地位に結果的に影響を与えるようなシナリオは避ける	例えば、AIシステムが個人の法的地位や法的権利、あるいは信用、教育、雇用、医療、住宅、保険、社会福祉給付、サービス、機会、あるいはそれらの提供条件へのアクセスに影響を及ぼす可能性があるシナリオが含まれます。	Azure OpenAI Service の透明性 Azure OpenAI Service の行動規範
危害を加える可能性のあるシナリオは避ける	Azure OpenAI サービスでホストされているモデルは、トレーニングデータやプロンプトで提供される例に存在する特定の社会的見解、偏見、その他の望ましくない内容を反映しています。そのため、不公平、信頼できない、または攻撃的な行動が及ぼす影響が大きい、または害につながる可能性のあるリスクが高いシナリオでモデルを使用しないように注意してください。	
利害関係の強い領域や業界におけるユースケースを慎重に検討する	例としては、ヘルスケア、医療、金融、法律などが挙げられます。これらに限定されるものではありません。	
チャットボットのシナリオを十分に検討する	チャットボットでのサービス利用を狭い領域に限定することで、意図しない反応や好ましくない反応が発生するリスクを低減することができます。	
すべての生成ユースケースを慎重に検討する	コンテンツ生成のシナリオは、意図しない出力を生み出す可能性が高く、このようなシナリオでは慎重な検討と緩和措置が必要です。	

Azure OpenAI 透明性シートの緩和策

Microsoft 365 Outlook Copilot の例



Human in the Loop

1c. 最終的な決定や最終的な内容について説明責任があることをユーザーに知らせる

1a. 生成された出力を編集できるようにする

1e. 生成されたコンテンツにおける AI の役割を開示

Azure OpenAI Service の透明性

Azure OpenAI Service のご利用

以下の申請フォームより利用用途とプロフィールの登録が必要



Azure OpenAI
Service

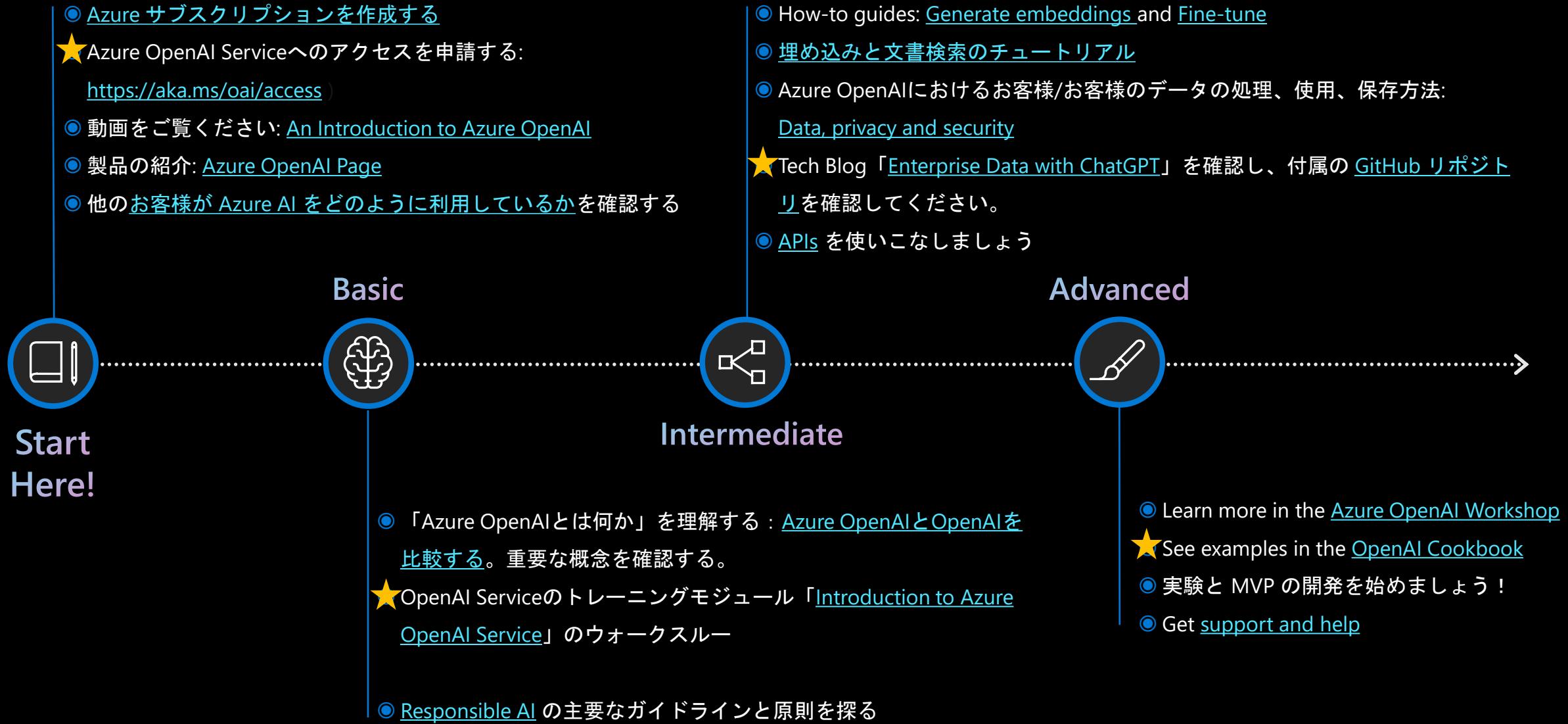
Azure OpenAI

<https://aka.ms/oaiapply>

GPT-4 (preview)

<https://aka.ms/oai/get-gpt4>

Azure OpenAI Service Learning Guide



Microsoft ❤ OSS

opensource.microsoft.com

[microsoft/semantic-kernel](https://github.com/microsoft/semantic-kernel)



[microsoft/DeepSpeed](https://github.com/microsoft/DeepSpeed)

 **DEEPSPEED CHAT** 

高速かつ低コストで学習

訓練時間 (hours)	費用 (Azure) (\$)
13B: 1.25hrs	\$320
175B: 22hrs	\$580

手軽に実行可能

- ChatGPTで用いられる RLHF訓練の完全な end-to-end/パイプラインをスクリプト一つで実行

高速・高スケーラビリティ

- ハイブリッドエンジンによって既存のRLHFシステムの15倍の高速化
- あらゆるスケールのモデルで類を見ないコスト削減を達成

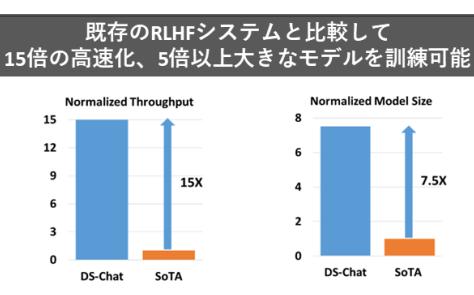
大規模モデルのサポート

- GPU1台で100億パラメータ超、複数GPUなら1000億パラメータ超のモデルを学習
- InstructGPT のパイプラインと様々な大規模モデルのファインチューニングをサポート

RLHFのための包括的な高速バックエンド

- ZeRO, LoRA等の技術を統合

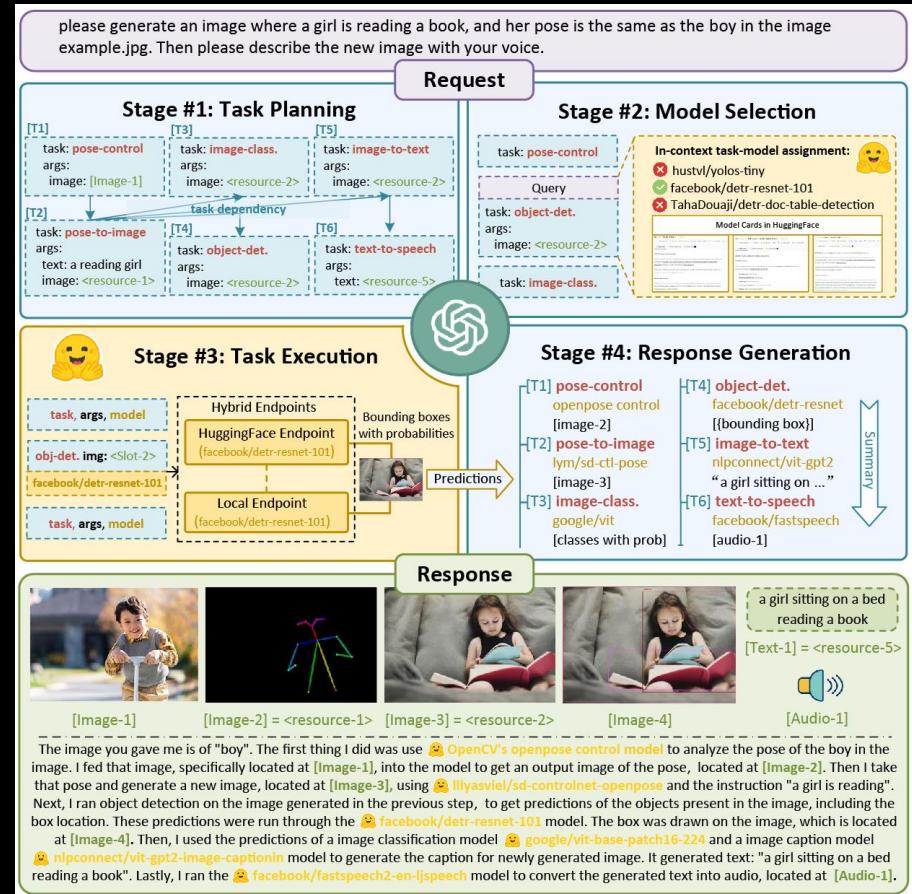
既存のRLHFシステムと比較して
15倍の高速化、5倍以上大きなモデルを訓練可能



Model	Normalized Throughput
DS-Chat	15
SoTA	1

Model	Normalized Model Size
DS-Chat	7.5X
SoTA	1

[microsoft/JARVIS](https://github.com/microsoft/JARVIS)
[microsoft/HuggingGPT](https://github.com/microsoft/HuggingGPT)



Partner Tech Blog with Azure OpenAI

Azure OpenAI Service – ナレコムAzureレシピ | AIに強い情報サイト (kc-cloud.jp)

- [Azure OpenAI活用ガイド: Azure OpenAIで責任ある開発のためのベストプラクティス](#)
- [APIキーも安全に！Azure DatabricksとAzure OpenAI Serviceの連携術](#)
- [連携で無限の可能性！Azure DatabricksとAzure OpenAI Service、10の革新的活用法](#)
- [効果的なデータ活用への道: Azure OpenAI ServiceとCosmos DBを組み合わせた企業内知識管理の実践](#)
- [送ったデータは使われちゃうの？よく聞かれるAzure OpenAI Serviceのデータプライバシーとセキュリティについてまとめました。](#)
- [Azure OpenAI サービスでマネージド ID を使った認証方法](#)
- [OpenAIとAzure OpenAI Serviceの比較: 10分で出来るOpenAIからのスムーズな移行ガイド](#)
- [TeamsにChatGPTを組み込むことで起こる業務変革4選](#)

JBS Tech Blog – AI Lab

- [Azure OpenAIにプライベートエンドポイント（Private Endpoint）を設定して東日本リージョンの仮想ネットワークのみから使う](#)
- [Azure OpenAI ChatGPTのAPIをPythonから叩いてみた](#)
- [Azure OpenAIのChatGPT APIとGPT-4を利用してFAQチャットボットを作成できた](#)
- [AzureのOpenAI\(GPT-3\)の埋め込みモデル\(Embeddings models\)を使ってQA検索の機能を比較してみた](#)
- [Azure OpenAI（GPT-3）と自動機械学習を組み合わせて自然言語からナレッジベースを構築する](#)
- [Azure OpenAIのChatGPTライクなAPIとAzure Cognitive Searchを連携して、FAQチャットボットのようにユーザーの質問に回答するシステムを構築できるか試してみた](#)

FIXER cloud.config blog

- [チームの生産性を向上！Azure OpenAI Serviceを使ってChatGPT\(GPT-4\)をSlackに召喚！](#)

ISID AI Transformation Center

- [AIと共に仕事をする時代に～ChatGPTの安心・安全な社内利用環境を作り、業務の生産性向上とAIに強い人材育成を加速～](#)

SB C&S Azure blog

- [Azure OpenAI Serviceとは？概要やChatGPTの利用についても解説](#)



パートナー・コミュニティ: Azure AI アップデート AI パートナーコミュニティのご紹介

- Teams の AI パートナーコミュニティ(JP-AIPC) を開設します
- 参加によって得られる情報
 - 最新の Azure AI アップデート情報の取得 (適宜)
 - パートナー向け AI セミナー/イベントの告知 (AI Update, ハッカソンなど)
 - セミナー等の資料のダウンロード
- 参加希望の方は jaaipc@microsoft.com までご連絡ください

Azure Applied AI Services



Azure Cognitive Services



Azure Machine Learning



Partners make



possible



年次最大の開発者会議

Microsoft Build



今年はオンラインと米国シアトルで開催!

オンライン(無料) : 日本時間 5月 24 日 (水) ~ 25 日 (木)

シアトル会場 : 米国時間(PDT) 5月 23 日 (火) ~ 25 日 (木)

最新情報・登録はQRコードまたは以下URLよりご確認ください
<https://build.microsoft.com>

日本の開発者様向けに、Microsoft Build Japan を
6月27日-28日に開催します (オンライン/東京会場)



Be there for the LIVE
keynotes

Be in the room as Satya Nadella, Chairman and CEO, shares key announcements, and learn from Microsoft's leading experts on Azure OpenAI Service, GitHub Copilot, GPT-4, .NET 8, and more.



Microsoft AI

