

投稿日 2025/03/28

利用者から提供者へ -AIエンジニア育成の取り組み-

AIエンジニア育成の取り組み

はじめに

生成AIの活用が進む中、単なるツール利用者ではなく「AIをシステムに組み込める」技術者の育成が急務となっています。TIS株式会社テクノロジー&イノベーション本部開発基盤センター（以下、開発基盤センター）では、実務で通用する生成AIエンジニア、特にRetrieval-Augmented Generation（以下、RAG）を扱える人材の育成に取り組んでいます。

この記事では、本AIエンジニア育成プログラムの狙いや構成、実践で得られた気づきや今後の展望をご紹介します。

背景と育成の狙い

TIS社内では、生成AIの技術検証やPoCを通じて、開発メンバーに知見が蓄積されつつあります。一方で、こうした取り組みをスケールさせるには大きな課題があります。それは、人材の偏在と不足です。

現在、生成AI開発に精通した人材は限られており、少人数の有識者にノウハウが集中しています。中でも、外部ドキュメントを活用してユーザーに適切な回答を返すRAGのニーズが高まっている一方で、これに対応できる人材は非常に限られています。

このような背景から、開発基盤センターにおいてはまずニーズの高いRAG分野にフォーカスし、**RAGエンジニア**を育成することを第一フェーズの目標としました。これは社内の技術者育成に向け、他の事業部門へ展開するための足がかりとも位置づけています。

RAGとは、外部データソース（業務文書など）から情報を取得し、生成プロセスに統合する手法であり、業務に根差したFAQやチャットボット構築に欠かせない技術です。とりわけ、検索精度・応答精度の改善には多くの技術的工夫と検証が必要であり、育成カリキュラムにおいてもこの「精度改善」に特に重点を置いて設計しています。

育成方針と対象者像

育成では以下の方針を掲げています。

- 実務に近い課題に対して、試行錯誤を通じた”手を動かす”学習

- 専門知識がない人でも参加できる、段階的な習得フロー
- 最終的にはRAGアプリを一から構築・評価できる力を身につける

受講者は若手社員が中心で、以下のようにバックグラウンドも多様です。

- 生成AIに初めて触れるエンジニア
- クラウドを触ったことがないインフラ未経験者
- 個人でChatGPTを使った経験があるが、開発は未経験の技術者

カリキュラム構成と学習内容

3か月の育成期間は以下のようなステップで構成されています。

フェーズ	学習テーマ	到達目標
0	環境構築	Python開発環境をローカルで構築できる
1	Python基礎とデータ処理	pandas等を使って実ファイルの前処理ができる
2	AWS上でのLLMデプロイ	Amazon SageMakerを使ったLLMのデプロイ・呼び出しができる
3	RAGの構築と精度改善	ドキュメントの前処理、インデックス作成、検索評価を実装できる
4	応答評価と実験管理	定量・定性的な応答評価が行える、ログの可視化ができる
5	模擬案件実践（Nablarch QA）	Javaフレームワーク（Nablarch）を対象としたRAG QAアプリを構築し、精度改善の取り組みとレポート作成ができる
6	市場調査と振り返り	トレンドを踏まえた活用シーンを調査・提案できる

各フェーズでは、受講者が実装ベースの課題に取り組み、Slackで進捗共有を行いながら自走学習を進めていきます。セクションごとに技術の前提知識を補足するガイドやTips集も用意しており、未経験者でもキャッチアップできるよう設計されています。

主としてインプットの学習が多いセクション1～セクション4の各セクションは、以下の表に示すとおりの子セクションで構成されており、それぞれの子セクションにおいてセクションゴールを設けることで、学習の到達点としています。また、セクションごとにステップアップコンテンツ（以下子セクション内の*印）を設けており、進捗状況に応じてより発展的な学習に取り組むことができるようにしています。またセクションの取り組み順については、学習マップで定義することで、進捗状況や個人の理解度に応じた学習を自発的に実施できるような仕組みを醸成しています。

■子セクションごとのゴール設定

セクション	タイトル	セクションゴール
-------	------	----------

セクション	タイトル	セクションゴール
1	Python基礎とデータ処理	
1.1	Pythonの基本	<ul style="list-style-type: none"> ・ Pythonで四則演算ができる ・ Pythonで文字列型の演算ができる ・ Pythonでif文やfor文の操作ができる ・ Pythonでリストや関数、クラスの取り扱いができる
1.2	Pythonライブラリの使い方	<ul style="list-style-type: none"> ・ csvファイルの読み書きができる ・ データの集計、整形、結合ができる ・ DataFrameを用いてテーブルデータから特定の列・行の抽出や演算・加工ができる ・ NumPyを用いて配列の計算ができる ・ Stremlitの外部ハンズオン講座を修了する
1.3	Pythonを用いたドキュメント処理	<ul style="list-style-type: none"> ・ 形態素解析について、参考資料を確認しながら自身の手で実行できる ・ Pythonにおけるスクレイピングについて、参考資料を確認しながら自身の手で実行できる ・ 欠損値データ補正について、参考資料を確認しながら自身の手で実行できる ・ doc2vecの概要を理解している ・ Pythonを用いたPDFファイルの操作方法について、参考資料を確認しながら自身の手で実行できる
1.4*	データ分析演習（詳細編）	<ul style="list-style-type: none"> ・ ハンズオン講座の修了
1.5*	機械学習（詳細編）	<ul style="list-style-type: none"> ・ 機械学習について、背景にどんな数学の知識が必要であり、それらがどう応用されているのか理解する
1.6*	Pythonを用いた機械学習のアルゴリズム実装	<ul style="list-style-type: none"> ・ ハンズオン講座の修了
1.7*	Pythonを用いた機械学習の数学的理解	<ul style="list-style-type: none"> ・ ハンズオン講座の修了
2	クラウド環境でのLLMデプロイと運用（AWS）	
2.1	AWS基礎編	<ul style="list-style-type: none"> ・ AWS環境へ接続する ・ AWSハンズオンセミナーを修了する
2.2	Amazon Bedrockの利用	<ul style="list-style-type: none"> ・ Amazon Bedrockのワークショップを修了する
2.3	AWSを利用した生成AIアプリケーション	<ul style="list-style-type: none"> ・ AWS Skill Builderで提供されている生成AIコンテンツ初級向けコンテンツを修了する ・ AWS Class Centralで提供されているAmazon Bedrock、Amazon Kendra、Amazon SageMaker、プロンプトエンジニアリングのコンテンツを修了する

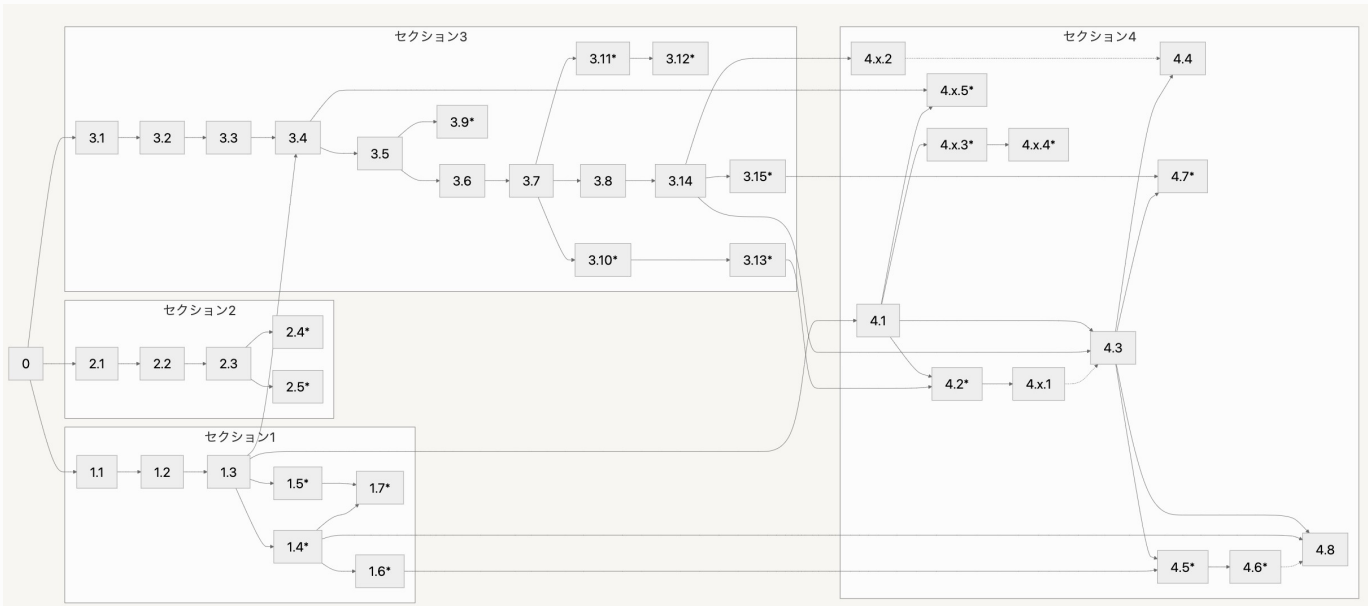
セクション	タイトル	セクションゴール
2.4*	Amazon Bedrock活用 (上級編)	・ Agents for Amazon Bedrock や Knowledge Base といったAmazon Bedrockの機能についてどのような機能であるか説明ができる
2.5*	AWSを用いた生成AIプロジェクトの遂行	・ ビジネスにおける生成AIの利用に関するAWS Class Centralのコンテンツを修了する ・ 学習内容について自身の言葉で説明ができる
3	RAGの理解と実装	
3.1	生成AI入門	・ 生成AIとは何か、そして大規模言語モデルの仕組みを理解する ・ 教育シナリオに重点を置き、さまざまなユースケースで大規模言語モデルを活用する方法を理解する
3.2	さまざまなLLMの調査と比較	・ ユースケースに適したモデルを選択する方法を理解する ・ モデルのパフォーマンスをテストし、反復し、そして改善する方法を理解する ・ モデルのデプロイ方法を理解する
3.3	生成AIの責任ある使用	・ 生成AIアプリケーション構築時の、責任あるAIの重要性について説明できる ・ 生成AIアプリケーションの構築時、責任あるAIの基本原則をいつ、どのように検討し適用するか説明できる ・ 責任あるAIの概念を実践するために利用可能なツールと戦略について説明できる
3.4	プロンプトエンジニアリングの基礎	・ プロンプトエンジニアリングとは何か、またその重要性について説明できる ・ プロンプトの構成要素とその使用方法について述べる ・ プロンプトエンジニアリングのベストプラクティスと技術を習得する ・ Amazon Bedrockのエンドポイントにアクセスし、学んだテクニックを実際に試す
3.5	高度なプロンプトの作成	・ 結果を改善するプロンプトエンジニアリング手法を実践する ・ 多様な、あるいは決定的なプロンプトを作成する方法を実践する
3.6	テキスト生成アプリケーションの構築	・ テキスト生成アプリケーションとは何かを説明できる ・ Amazon Bedrockを使用してテキスト生成アプリを構築する ・ 使用するトークン数を増減させたり、温度を変更し、さまざまな出力が得られるようにアプリを構成する
3.7	生成AIを利用したチャットアプリケーションの構築	・ チャットアプリケーションを実装し、既存のシステムに統合する際の考慮点を理解する ・ 特定のユースケースに合わせたチャットアプリケーションのカスタマイズ方法を理解する ・ AIを活用したチャットアプリケーションの品質を効果的に監視し、維持するために重要な指標と考慮点を特定する ・ 責任あるAIの原則をチャットアプリケーションで適用する
3.8	検索アプリケーションの構築	・ セマンティック検索とキーワード検索の違いを理解し、説明できる ・ テキスト埋め込みとは何かを理解し、説明できる

セクション	タイトル	セクションゴール
		<ul style="list-style-type: none"> ・埋め込みを使用した検索アプリケーションを作成できる
3.9*	画像生成アプリケーションの構築	<ul style="list-style-type: none"> ・画像生成アプリを作成する ・メタプロンプトを用いてアプリケーションの範囲を定義する ・DALL-EとMidjourneyの活用方法を理解する
3.10*	関数呼び出しとの統合	<ul style="list-style-type: none"> ・関数呼び出しを使う理由を説明できる ・Amazon Bedrockを使用して関数呼び出しをするアプリを構築する ・アプリケーションのユースケースに適した効果的な関数呼び出しを設計する
3.11*	AIアプリのUX設計	<ul style="list-style-type: none"> ・ユーザーのニーズを満たすAIアプリを構築する ・信頼性とコラボレーションを促進するAIアプリの設計方法を理解する
3.12*	生成AIアプリケーションのセキュリティ	<ul style="list-style-type: none"> ・AIシステムに対する脅威とリスクを理解する ・AIシステムを守るための一般的な方法とプラクティスを理解する ・セキュリティテストの実施により、予期せぬ結果とユーザーの信頼低下を防ぐことができる
3.13*	汎用AIアプリケーションのライフサイクル	<ul style="list-style-type: none"> ・MLOpsからLLMOpsへのパラダイムシフトについて説明できる ・LLMのライフサイクルについて説明できる ・ライフサイクルツーリングについて説明できる ・ライフサイクルの計測と評価について説明できる
3.14	検索拡張生成(RAG)とベクトルデータベース	<ul style="list-style-type: none"> ・RAGのデータ検索・処理における重要性を説明できる ・RAGアプリケーションを設定し、LLMにデータを組み込める ・LLMアプリケーションにRAGとベクトルデータベースを効果的に統合できる
3.15*	AIエージェント	<ul style="list-style-type: none"> ・AIエージェントが何か、どのように使えるかを説明できる ・人気のあるAIエージェントフレームワークの違いを理解する ・AIエージェントの機能を理解し、アプリケーション構築に活用できる
4	LLMアプリの品質評価と実験管理	

セクション	タイトル	セクションゴール
4.1	入門LangSmith	<ul style="list-style-type: none"> LangSmithのAPIキーを発行しAPI呼び出しができる LangSmithを用いてトレースが記録できる システムの性能を定量的に測定できる
4.2*	LLMアプリケーションのオブザーバビリティ	<ul style="list-style-type: none"> オブザーバビリティについて説明できる オブザーバビリティを持つLLMアプリケーションをセットアップできる
4.3	LLMアプリケーションの評価	<ul style="list-style-type: none"> 性能測定に使用するテスト用データセットの作成ができる 性能測定に使用する指標が定義できる 異なるプロンプトやモデルを用いて性能評価ができる 時系列での結果トレースができる
4.4	RAGの評価	<ul style="list-style-type: none"> RAGの評価についてそのフローを理解し、実施内容を説明できる RAGの評価について複数の手法を用いた評価ができる
4.5*	分類器で学ぶ出力の最適化	<ul style="list-style-type: none"> 分類器とは何かについて説明できる 分類器によるオートメーションを構築し、フィードバックの自動分類ができる
4.6*	バックテスト	<ul style="list-style-type: none"> バックテストとは何かについて説明できる サンプリングされたデータを用いて他のシステムの性能測定ができる
4.7*	エージェントの評価	<ul style="list-style-type: none"> LLMエージェントの評価手法についての理解があり、実践できる
4.8	センチメント分析	<p>センチメント分析をテーマに以下のタスクを完了させる</p> <ul style="list-style-type: none"> データセット準備 モデルの構築・評価 モデルの分類性能向上
4.x.1	トレーシングに関わる概念	<ul style="list-style-type: none"> トレーシングに関わる概念の理解がある
4.x.2	Ragasに学ぶRAGの評価指標	<ul style="list-style-type: none"> RAGの評価指標の理解がある
4.x.3*	データセットの管理	<ul style="list-style-type: none"> データセットの管理の理解がある

セクション	タイトル	セクションゴール
4.x.4*	データセットのバージョン管理	・ データセットのバージョン管理の理解がある
4.x.5*	プロンプトの管理	・ プロンプトの管理の理解がある

■学習マップ



実践を通じた学びと気づき

育成プログラムを通じて、以下のような気づきがありました。

精度改善の難しさと面白さ

多くの受講者が「精度評価」という新しい観点に苦戦しながらも、最も成長を実感した領域でした。特に、どのようなチャンキング手法が有効か、検索クエリの最適化がどれだけ応答に影響するかなどを自ら検証し、試行錯誤を重ねる体験は、明確な仕様と正解がある従来型のシステム開発とは異なり、“より良い応答”を模索する相対的な最適解を目指す、生成AIならではの開発体験です。

検索精度が悪い理由を分析し、チャンキングの粒度を変えるだけで大きく改善されたときは面白かった
(受講者コメント)

プロンプト設計 → 評価 → 改善 という試行錯誤の繰り返しは、要件が明確に定義された仕様駆動型の開発とは異なり、出力の揺らぎや正解が一つに定まらない中で最適解を模索していくプロセスです。データの非決定性や相対的な評価基準と向き合うことで、受講者は柔軟で探索的な技術的思考を身につけています。

全体を通じた”変化”

受講前は「生成AIって難しそう」と感じていたメンバーも、育成後には「まず試してみる」「プロンプトを調整してみる」という姿勢に変わりました。自らAIを動かすことへの心理的障壁が下がったことも大きな成果です。

実際の受講者成果物（一例）

以下は実際の受講者成果物の一例です。RAGシステムの応答精度について定義後、試行錯誤で応答精度改善に取り組み、実際に定量的な精度を向上させることができています。

ここで紹介する応答評価は、RAGの性能を定量的に測定するために使用される指標であり、Ragasという評価フレームワークを用いています。Ragasは、RAGパイプライン内の各プロセス（検索・文脈選定・応答生成）に対して個別に評価指標を定めており、具体的には以下ような観点でスコアリングを行います：

- **Faithfulness（忠実度）**：生成された回答から特定される主張が与えられたコンテキストから推測できるかどうか
- **Answer Relevancy（回答の関連度）**：回答が質問に対してどれだけ関連しているか
- **Context Recall（コンテキストの再現率）**：検索されたコンテキストが、正解として扱われる注釈付き回答との程度一致するか
- **Answer Correctness（回答の正しさ）**：生成された回答と正解の間の整合性が高いかどうか

#	Faithfulness	Answer Relevancy	Context Recall	Answer Correctness
改善前平均	0.766	0.746	0.812	0.634
改善後平均	0.883	0.847	0.935	0.667

これらの指標値の改善手法として、RAGの精度改善に関する取り組みを実施しています。今回の例においては具体的には以下のような改善策を取り入れることで、精度向上を成し遂げることができました。

- ・ 回答が質問の意図に沿うようにLLMへのクエリに対するプロンプトエンジニアリングの実施
- ・ RAGのドキュメント検索精度向上のため、チャンキング戦略の改善
- ・ より期待する回答に近付けるため、回答結果に応じた回答の再作成プロセスの構築

今後の展望

今後は次の2つの軸で育成の幅を広げていきます。

1. スピードアップと対象者の拡大

現状は3か月×3名のサイクルで実施していますが、育成スピードと対象者数を増やすために、以下のような拡張を検討中です：

- コンテンツのオンデマンド化
- 社内の事業部へのコンテンツ展開
- 現場でのRAG導入支援へ繋がる動線の形成

2. AIインテグレーション人材の育成

RAGに限らず、AIエージェントや多段階プロンプト、社内DBとの統合など「AIをどうシステムに組み込むか」に踏み込んだAIインテグレーション人材育成も構想中です。

私たちが目指しているのは、「すべてを一人で担えるスーパーマン」を育てることはありません。RAGエンジニア育成を含む本取り組みのゴールは、AI技術を業務の中で適切に活用し、自ら考えて改善できる、そんな“第一歩”を踏み出せる人材を一人でも多く育てることにあります。

まとめ

本育成プログラムでは、「RAGを理解する」だけに留まらず、精度改善という実践的で奥深いテーマに真正面から向き合うことで、既存のSI開発とは異なる経験と視点を育んでいます。

受講者たちは、模索と試行錯誤の中で生成AIとの向き合い方を学び、自らの手でAIを組み込む実感を得ました。これは、単なる技術研修ではなく、“生成AI時代のエンジニア像”を自ら体感するプロセスだと言えるでしょう。

「AIを活用できる」から、「AIを開発し、改善できる」エンジニアへ。

そんな未来を描く第一歩を、私たちは踏み出しています。

/* Recommend */

「Generative AI（生成AI）」のおすすめ記事はこちら


この記事に関連する記事もお読みください。

 Generative AI（生成AI）


NEW


OctoNihon Forumイベント発表資料「GitHub Copilotを活用した大規模開発の今と未来」公開

2025/06/06  4


 Generative AI（生成AI）

96%が満足！「わかる」AIエージェント実践型ワークショップ


2024/12/20  20

 Generative AI（生成AI）


社内AIチャット：『TIS AIChatLab』の技術スタック公開

2024/11/08  42

最近投稿された記事も用意しました。

 Generative AI（生成AI）**NEW**

「金融業界特化型AIエージェント
ワークショップ」開催レポート

2025/06/13  8


 Generative AI（生成AI）**NEW**

OctoNihon Forumイベント発表
資料「GitHub Copilotを活用した
大規模開発の 今 と 未来」公開

2025/06/06  4

ITアーキテクチャ**NEW**

生成AIの新潮流：AIエージェント
勉強会を開催しました

2025/05/02  12

「Generative AI（生成AI）」で最も読まれている記事を以下にまとめています。

 Generative AI（生成AI）

GitHub Copilotを用いたAIによる
コードレビューの活用

2025/01/28  26

 Generative AI（生成AI）**NEW**

OctoNihon Forumイベント発表
資料「GitHub Copilotを活用した
大規模開発の 今 と 未来」公開

2025/06/06  4

 Generative AI（生成AI）

GitHub Copilotを活用した大規模
開発～オフショア開発での実践と
知見～

2025/02/26  16