Spectrum Technology

V1.0 2023/7/24

抜粋

はじめてのchatGPT開発キット

RaspberryPiでchatGPTの基礎から関連APIで応用まで習得、体験

応用編

ChatGPT

LangChain

LlamaIndex

Raspberry Pi

スペクトラム・テクノロジー株式会社

https://spectrum-tech.co.jp

sales1@spectrum-tech.co.jp

Spectrum Technology

開発キット　目次

	ページ
• 開発キット　全体像	4
• ハード概要	5
• ソフト概要	6
1.　Chatgpt連携	
①　ChatGPTと関連api	7
2.　Langchain	
①　Langchain概要	8
②　Langchain機能一覧	9
③　事例	
A)　Autonomous (long-running) agents	10
B)　Agent simulations	18
C)　Agent	26
D)　Chatbot	33
E)　Code Understanding	34
F)　multi modal	36
G)　Question answering	37
H)　Evaluation	38

抜粋版のためページと一致しません

開発キット　目次

3. LlamaIndex
   - ① LlamaIndex概要 46
   - ② LlamaIndex機能一覧 47
   - ③ 事例
     - A) Agent 48
     - B) Customization 51
     - C) chat engine 52
     - D) data_connectors 54
     - E) Index 56
     - F) Query Engine 60
     - G) Analysis 65
     - H) output_parsing 67
     - I) Evaluation 70
     - J) Integrations 71
     - K) Callbacks 75
4. AI detector 78
5. Stable Diffusion 81

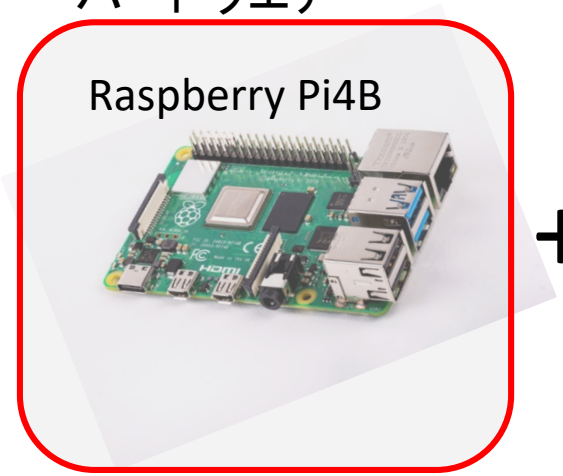
抜粋版のためページと一致しません

Spectrum Technology

開発キット　全体像(Pi版)

ハードウエア

Raspberry Pi4B



＋

物品追加

電源アダプタ
（オプション）

HDMIケーブル
（オプション）

OS

Rasbian OS

API関系

ChatGPT

LangChain

LlamaIndex

画像系

Stable diffusion

（オプション）

プログラム言語

python

Triton(オプション)

AI detector

GPTZero

crossplag

Copyleaks

Spectrum Technology

ハード概要

本体

品名	項目	内容	備考
Raspberry Pi 4 model B	CPU	1.5GHz 64bit クアッドコアCoretex-A72	
	GPU	デュアルコア VideoCore VI® 500MHz	
	メモリ	4GB RAM	
	OS	Raspbian bullseye(Debianベース)	
	インターフェース	2.4/5GHz WiFi(802.11 bgnac), Bluetooth 5.0, BLE, 1G ether, USB 2.0x2, USB 3.0x2, micro HDMIx2, microSDカード, 40 GPIO pin	
	電源／消費電力	Micro USB Type C 3.0A	
	サイズ	85x56x18mm	

付属品	内容	備考
ケース	透明、白、黒、銀から選択できます。	
microSD 32GB	Raspbian OS, 必要なモジュールをインストールして提供します。お客様が設定するものは必要最低限のパスワード設定、WiFi設定になります。	
プログラム	Chatgpt,langchain,llamaindex, 関連pip インストール済, Pythonサンプルプログラム多数	
マニュアル	開発キット 基本編、応用編	

USB電源ケーブル、HDMIケーブルは付属しておりません。
別途オプション品を購入ください

ソフト概要

提供するソフトウエアの概要です。

区分	ソフト名	バージョン	備考
OS	Raspbian	Bullseye 64bit	
プログラム言語	python	3.9.2	
APIプログラム	openai	0.27.8	
	langchain	0.0.201	
	llamaindex	0.6.28	
その他	Jupyter notebook、matplotlibなど多数のpipライブラリ		

1. ChatGPT連携
①. ChatGPTと関連api

- langchain
  - Chatgptは、2021年9月までのデータを学習している。それ以降のデータを使用するためlangchainを使用。Webのgoogle検索データを出力します。
  - https://langchain.com/
  - 用途：chatbot, 質疑応答
  - 主な特徴
    - 2021年10月以降のデータを出力
    - 最大の入出力合計が4096トークンであるため、長い情報を持たせる
    - ExcelやCSV、PDF等を直接読み込ませる
    - 会話の履歴管理
- LlamaIndex(旧gpt index)
  - LlamaIndexは大規模言語モデル(LLM)と外部データ(あなた自身のデータ)を接続するためのインターフェースを提供
  - https://gpt-index.readthedocs.io/en/latest/
  - 用途：・質疑応答・要約・テキスト生成
  - 主な特徴
    - LLMに外部情報を受け渡すための構造化データを作成する
    - 作成した構造化データを踏まえて質問に回答するようLLMに要求する処理を実現する
- Ai detector
  - コンテンツが生成AIからの出力によるものかの判定。クラウドベースのサービス。ほとんどが有料。
  - 各クラウドサービスを紹介
- Stable Diffusion
  - 画像生成AI，openaiのDALL·E2と競合、オープンソースのため無料で利用可能。但し、gpu搭載の高スペックpcが必要（RTX3080でもメモリ不足で動かない）
  - 別途インストールサービスを提供

2. Langchain
①. Langchain概要

- Langchainとは
  - LangChain は、言語モデルを利用してアプリケーションを開発するためのフレームワークです。これにより、次のようなアプリケーションが可能になります。
    - データ認識: chatgptのような言語モデルを他のデータソースに接続する
    - エージェントティック: 言語モデルがその環境と対話できるようにします。
  - 特にchatgptの欠点を補完します。
    1. 最新情報に対応していない。
    2. 長文を入力できない。
    3. 複雑な計算問題に回答できない。
  - https://langchain.com/
- 機能
  - Model I/O:複数のモデルと連携。Chatgpt, seapapi,hugging face
  - Prompts:プロンプトのテンプレート化やプロンプトの構築と操作を容易にするいくつかのクラスと関数を提供します。
  - Chains：複数のプロンプト入力を実行する機能
  - Agents:
    - アクション エージェント: 各タイムステップで、以前のすべてのアクションの出力を使用して次のアクションを決定します。
    - エージェントの計画と実行: 一連のアクションを事前に決定し、計画を更新せずにすべてを実行します。
  - Memory: ChainsやAgentsの内部における状態保持をする機能
  - Evaluation:出力結果を複数の評価指標で真偽を評価します。

2. Langchain

②. Langchain機能一覧

項番	項目	api名等	内容	方式	備考
1	Autonomous (long-running) agents	baby_agi	BabyAGI を LLM チェーンとして実装するノートブック	ipynb	SFの今日の天気
2		baby_agi_with_agent	BabyAGI を LLM チェーンとして実装するノートブック,serpapi使用	ipynb	SFの今日の天気
3		baby_agi_with_agent2	BabyAGI を LLM チェーンとして実装するノートブック,google cutom search使用の場合	ipynb	SFの今日の天気
4		AutoGPT	2者間での自動的に会話	ipynb	SFの今日の天気
5		MetaPrompt	同じような質問を繰り返す	ipynb	
6		marathon_times	2018年から2022年のボストンマラソンの優勝者とタイム問い合わせ	ipynb	
7	Agent simulations	camel_role_playing	自律型の会話モデル,プログラマとトレーダで株取引の開発システムを構築する手順	ipynb	
8		Characters	人物設定をして、キャラクタと会話	ipynb	
9		Gymnasium	agentの環境を設定できる	ipynb	
10		multi_player_dnd	複数のagent設定し、会話、Harry Potter、Dungeon Masterとの会話	ipynb	
11		multiagent_authoritarian	複数のagent設定し、会話、特に会話前に考え、会話も終了させる。4名のタレントの会話	ipynb	
12		multiagent_bidding	複数のagent, topic設定し会話、debate形式。bidして発言、["Donald Trump", "Kanye West", "Elizabeth Warren"]	ipynb	
13		petting_zoo	ジャンケン、カードゲームを行うagent	ipynb	
14		two_player_dnd	二人のagentによる会話,Harry Potter、Dungeon Masterとの会話	ipynb	
15	Agent	baby_agi	FAISS(facebook ai research)のモデルを使った会話	ipynb	SFの今日の天気
16		baby_agi_with_agent	FAISS(facebook ai research)のモデルを使った会話	ipynb	SFの今日の天気
17		camel_role_playing	2名によるロールプレイ,プログラマとトレーダで株取引の開発システムを構築する手順	ipynb	
18		custom_agent_with_plugin_retrieval	pluginを伴ったagentの会話	ipynb	
19		multi_modal_output_agent	マルチモーダル(textから画像生成)のagent、有料なので中断	ipynb	有料
20		sales_agent_with_context	sales agentとの会話	ipynb	
21		wikibase_agent	wikiとの会話	ipynb	デフォルトは、gpt-4
22	chatbot	voice_assistant	音声認識による入力と会話	ipynb	piではメモリ不足
23	Code Understanding	code-analysis-deeplake	Langchain + Activeloop 's Deep Lake with GPTの分析	ipynb	エラー中不明
24		twitter-the-algorithm-analysis-deeplake	Langchain + Activeloop 's Deep Lake with GPTを使ってtwitterを分析	ipynb	
25	multi modal	image_agent	テキストからstemship経由でdall-eを使って画像生成	ipynb	有料
26	Question answering	semantic-search-over-chat	Langchain + Activeloop 's Deep Lake with GPTを使って検索	ipynb	
27	evaluation	agent_benchmarking	Serpapiなどを使って予測の評価を実施	ipynb	
28		data_augmented_question_answering	Serpapiなどを使って予測の評価を実施	ipynb	
29		huggingface_datasets	huggingface_datasets使って予測の評価を実施	ipynb	
30		llm_math	llm_math使って数字の期待値と予測を実施	ipynb	
31		openapi_eval	openapi使って予測と評価を実施	ipynb	
32		qa_benchmarking_pg	Paul Graham Essay使って評価を実施	ipynb	
33		qa_generation	qa_generation_chain使って回答	ipynb	
34		question_answering	複数ある評価方法を比較	ipynb	

2. Langchain
③事例
A). Autonomous (long-running) agents

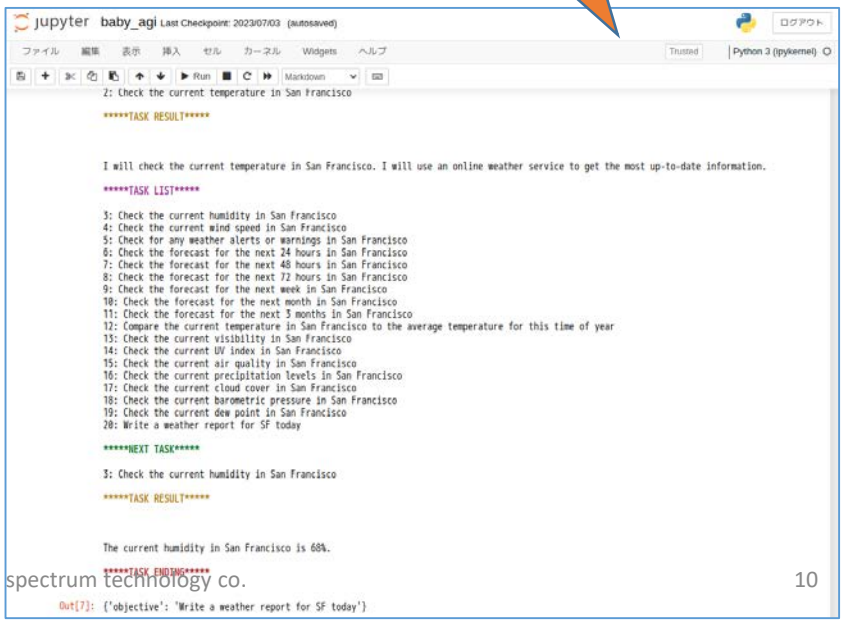
コマンド入力
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook

- トップページ https://python.langchain.com/docs/get_started/introduction
- Github https://github.com/hwchase17/langchain
- Autonomous (long-running) agents
  - 自律エージェントは、より長時間実行されるように設計されたエージェントです。彼らに１つまたは複数の長期目標を与えると、彼らはその目標に向かって自主的に実行します。アプリケーションは、ツールの使用と長期記憶を組み合わせます。
  - baby_agi
  - BabyAGI を LLM チェーンとして実装するノートブック

$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook
Autonomous_Agents>baby_agi.ipynb

Pipのインストール部分はskipして進んでください。インストール済。

今日のSan franciscoの天気？



2. Langchain

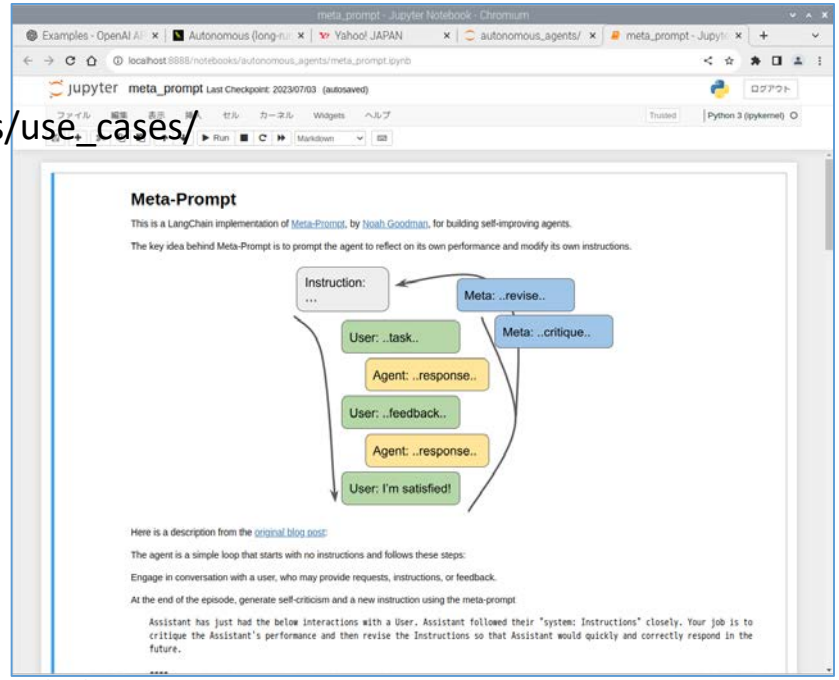
③事例

A). Autonomous (long-running) agents

コマンド入力
$ export SERPAPI_API_KEY=‘xxx'
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook

- トップページ https://python.langchain.com/docs/get_started/introduction
- Github https://github.com/hwchase17/langchain
- Autonomous (long-running) agents
  - 自律エージェントは、より長時間実行されるように設計されたエージェントです。彼らに１つまたは複数の長期目標を与えると、彼らはその目標に向かって自主的に実行します。アプリケーションは、ツールの使用と長期記憶を組み合わせます。
  - MetaPrompt
  - 同じような質問を繰り返す

$ export SERPAPI_API_KEY=‘xxx'
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook
Autonomous_Agents>meta_prompt.ipynb



2. Langchain
③事例
C). Agent

コマンド入力
$ export STEAMSHIP_API_KEY=‘xxx’
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook

- トップページ https://python.langchain.com/docs/get_started/introduction
- Github https://github.com/hwchase17/langchain
- Agent
  - エージェントはさまざまなタスクに使用できます。エージェントは、言語モデルの意思決定能力とツールを組み合わせて、ユーザーに代わってソリューションを実行および実装できるシステムを作成します。
  - multi_modal_output_agent
  - マルチモーダル（textから画像生成）のagent、有料なので中断。

$ export STEAMSHIP_API_KEY=‘xxx’
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/langchain/docs/use_cases/
$ jupyter notebook
Agent>multi_modal_output_agent.ipynb

Steamshipのapiを使用。月額＄10かかるので中断

3. LlamaIndex
①. LlamaIndex概要

- LlamaIndexとは
  - LlamaIndexは大規模言語モデル(LLM)と外部データ(あなた自身のデータ)を接続するためのインターフェースを提供
  - https://gpt-index.readthedocs.io/en/latest/
  - 用途：・質疑応答・要約・テキスト生成
  - 主な特徴
    - LLMに外部情報を受け渡すための構造化データを作成する
    - 作成した構造化データを踏まえて質問に回答するようLLMに要求する処理を実現する
- 機能
  - agent：自動化された推論および意思決定エンジンです。
  - Customization:chatgptに自社データ取り込み
  - chat engine：チャット用のエンジン
  - data_connectors： LlamaHubを使って接続
  - Index：インデックスは、ユーザー クエリに関連するコンテキストを迅速に取得できるようにするデータ構造です。
  - Query Engine：自社データへの質問を可能にするエンジン

3. LlamaIndex

②. LlamaIndex機能一覧

項番	項目	api名等	内容	方式	備考
1	agent	openai_agent	gpt-3.5のopenaiモデルのagent,掛け算の例	ipynb	
2		openai_agent_with_query_engine	query_engine toolを使って、uber, lyftの会社情報にも続いて回答。元は、決算情報のpdfからindex作成	ipynb	
3		openai_agent_retrieval	掛け算などの関数を定義して、その関数を呼び出して、openaiが回答	ipynb	
4	customization	SimpleIndexDemo-ChatGPT	paul_grahamのドキュメントを読み込んで、LLM Predictor (gpt-3.5-turbo) : default とChatGPTLLMPredictorとの差分比較	ipynb	
5	chat engine	chat_engine_condense_question	その後どうなったなどの前の質問をフォローする。paul_grahamのエッセイに対する複数回答	ipynb	
6		chat_engine_repl	davinci, gpt-3.5の比較	ipynb	
7	data_connectors	DeepLakeReader	Activeloopのエッセイに接続し質問。paul_grahamのエッセイに対する回答	ipynb	
8		WebPageDem	http://paulgraham.com/worked.html のサイトに接続して質問	ipynb	
9	index	DocSummary	wikiのwiki_titles = ["Toronto", "Seattle", "Chicago", "Boston", "Houston"]の都市データを読み込み、概要を表示	ipynb	
10		PandasIndexDemo	["Toronto", "Tokyo", "Berlin"]の都市データを読み込み、概要を表示	ipynb	
11		SQLIndexDemo	["Toronto", "Tokyo", "Berlin"]の都市データをsqlデータで読み込み、概要を表示	ipynb	
12		KnowledgeGraphDemo	データをグラフ化して、可視化	ipynb	
13	Query Engine	json_query_engine	json形式のデータを読み込み、そのクエリー	ipynb	
14		sub_question_query_engine	複数のデータに対する回答。複数からの回答を合成.paul_grahamのエッセイに対する表示	ipynb	
15		SQLRouterQueryEngine	都市データをsqlで読み込み設定し、wikiの情報も取り込み回答する	ipynb	
16		citation_query_engine	引用データの大きさ変更することもできる.paul_grahamのエッセイに対する表示	ipynb	
17		CustomRetrievers	and, orを使ってdataを使用	ipynb	
18	Analysis	PlaygroundDemo	wikiのberlinのデータを読み込んでplaygroundによる会話	ipynb	
19		TokenPredictor	indexの違いによる、tokens数算出	ipynb	
20	output_parsing	GuardrailsDemo	Guardrails形式の出力例	ipynb	
21		LangchainOutputParserDemo	json形式の出力例	ipynb	
22		guidance_pydantic_program	json形式の出力例	ipynb	
23	Evaluation	RetryQuery	Retryqueryによる評価	ipynb	
24	Integrations	SimpleIndexDemo	簡単なベクトル保存場所のデモ	ipynb	
25		QdrantIndexDemo	Qdrantのベクトル保存場所のデモ	ipynb	
26		DeepLakeIndexDemo	DeepLakeのベクトル保存場所のデモ	ipynb	
27		ChromaIndexDemo	ChromaDBのベクトル保存場所のデモ	ipynb	
28	callbacks	TokenCountingHandler	トークン数算出	ipynb	
29		LlamaDebugHandler	レスポンスタイムなどのデバッグ機能	ipynb	
30		AimCallback	インプットとアウトプットのトレース機能	ipynb	

3. LlamaIndex

③事例

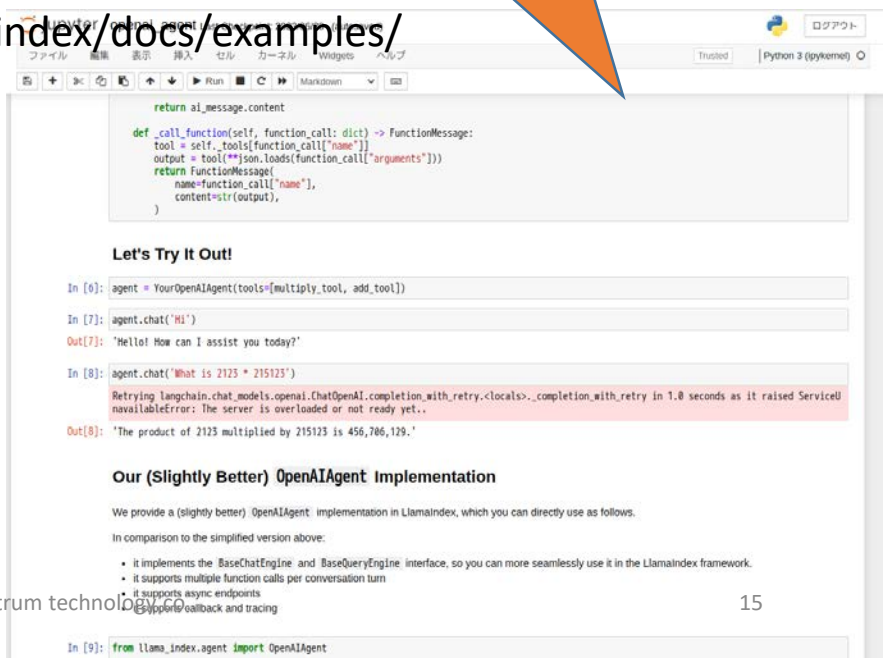
A). agent

コマンド入力
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook

- トップページ https://gpt-index.readthedocs.io/en/latest/
- Github https://github.com/jerryjliu/llama_index
- Agent
  - 「エージェント」は、自動化された推論および意思決定エンジンです。ユーザー入力/クエリを受け取り、正しい結果を返すためにそのクエリを実行するための内部決定を行うことができます。
  - openai_agent
  - gpt-3.5のopenaiモデルのagent。

$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook
agent>openai_agent.ipynb

Pipのインストール部分はskipして進んでください。インストール済。



掛け算の例

3. LlamaIndex

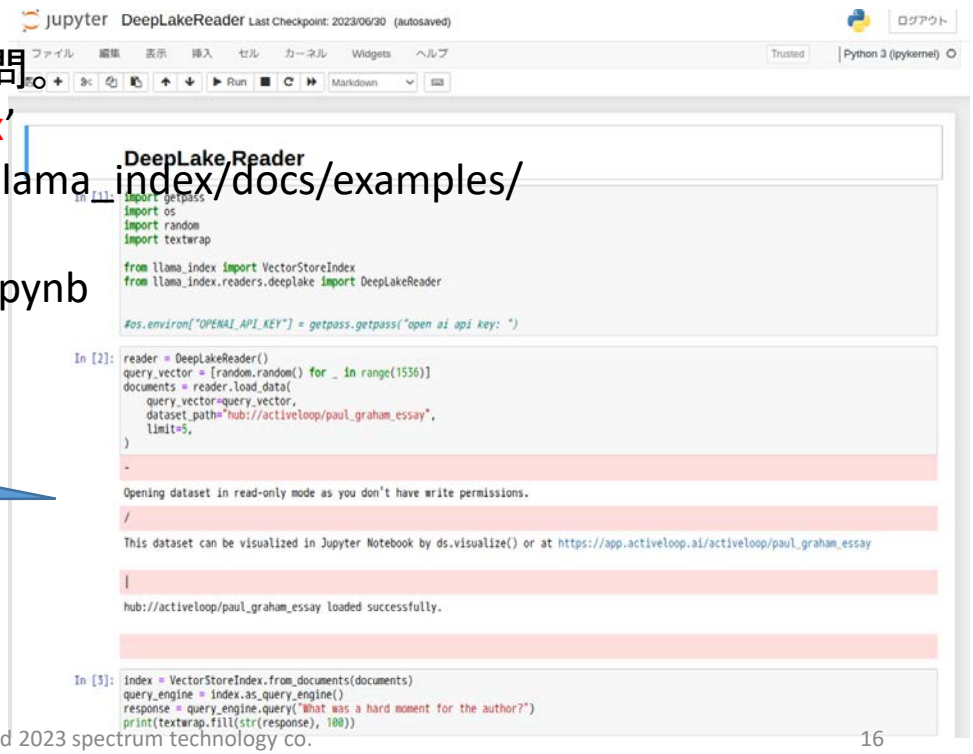
③事例

D). data_connectors

コマンド入力
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook

- トップページ https://gpt-index.readthedocs.io/en/latest/
- Github https://github.com/jerryjliu/llama_index
- data_connectors
  - LlamaHub は、任意の LlamaIndex アプリケーションに簡単にプラグアンドプレイできるデータローダーを含むオープンソースリポジトリです。。
  - DeepLakeReader
  - Activeloopのエッセイに接続し質問。

$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook
data_connectors>DeepLakeReader.ipynb



paul_grahamのエッセイに対する回答

3. LlamaIndex

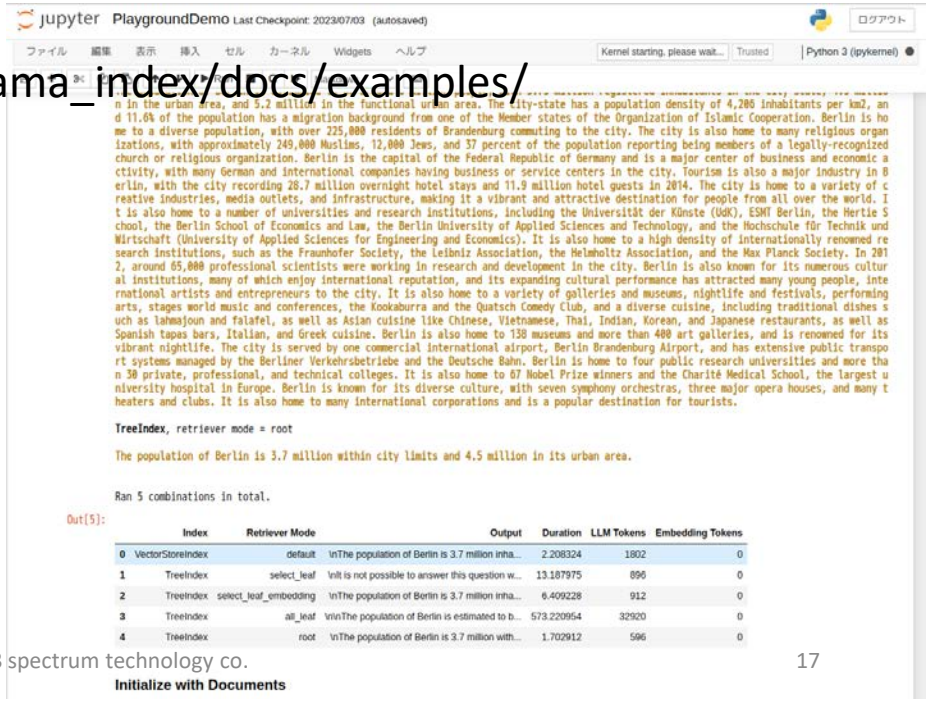
③事例

G). Analysis

- トップページ https://gpt-index.readthedocs.io/en/latest/
- Github https://github.com/jerryjliu/llama_index
- Analysis
  - tokens数によるコスト分析など。
  - PlaygroundDemo
  - wikiのberlinのデータを読み込んでplaygroundによる会話。

$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook
Analysis>PlaygroundDemo.ipynb

コマンド入力
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook



	Index	Retriever Mode	Output	Duration	LLM Tokens	Embedding Tokens
0	VectorStoreIndex	default	\nThe population of Berlin is 3.7 million inha...	2.208324	1802	0
1	TreeIndex	select_leaf	\nIt is not possible to answer this question w...	13.187975	896	0
2	TreeIndex	select_leaf_embedding	\nThe population of Berlin is 3.7 million inha...	6.409228	912	0
3	TreeIndex	all_leaf	\n\nThe population of Berlin is estimated to b...	573.220954	32920	0
4	TreeIndex	root	\nThe population of Berlin is 3.7 million with...	1.702912	596	0

Spectrum Technology

3. LlamaIndex

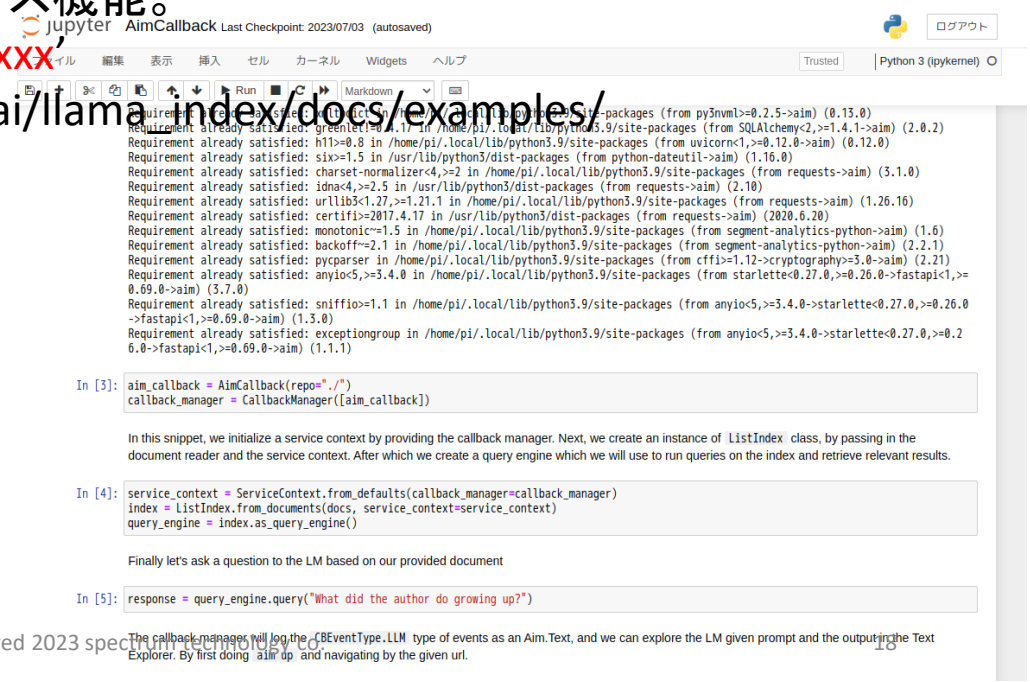
③事例

K). callbacks

コマンド入力
$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook

- トップページ https://gpt-index.readthedocs.io/en/latest/
- Github https://github.com/jerryjliu/llama_index
- callbacks
  - ライブラリの内部動作のデバッグ、追跡、トレースに役立つコールバックを提供します。
  - AimCallback
  - インプットとアウトプットのトレース機能。

$ export OPENAI_API_KEY=‘sk-xxxxx’
$ cd /home/pi/Documents/openai/llama_index/docs/examples/
$ jupyter notebook
callbacks>AimCallback.ipynb

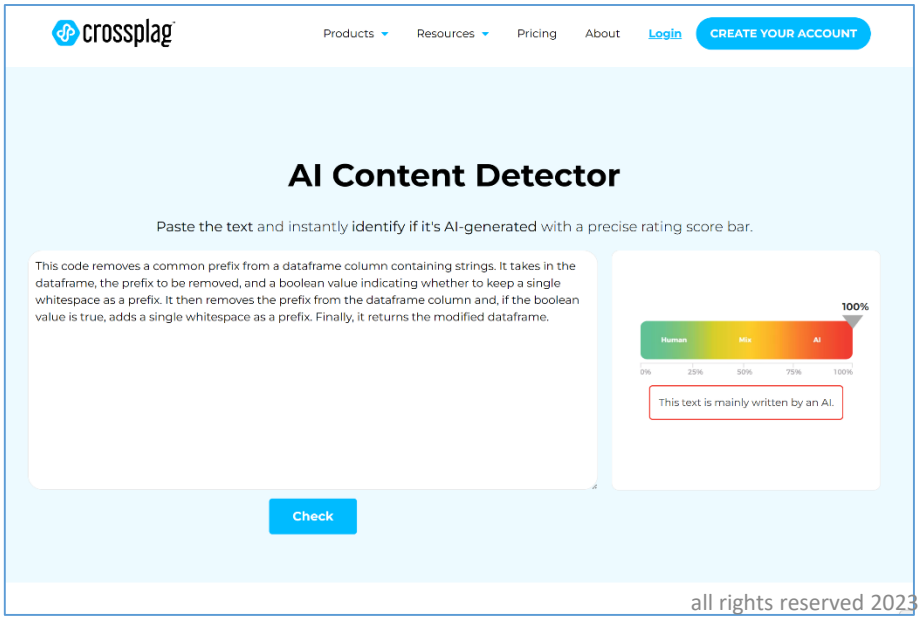


4. AI detector
②. AI detector比較

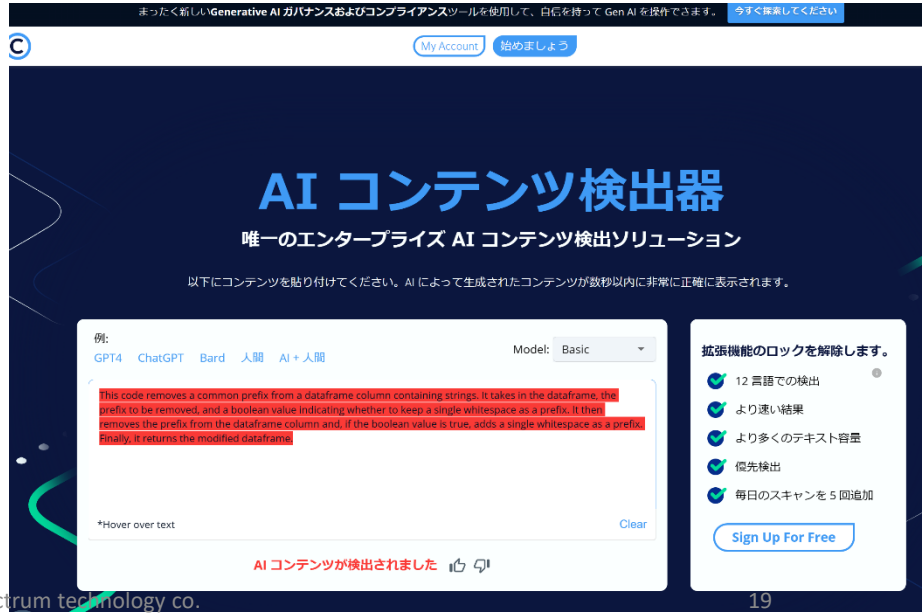
4.Openai事例⑪. Python to natural languageの出力
This code removes a common prefix from a dataframe column containing strings. It takes in the dataframe, the prefix to be removed, and a boolean value indicating whether to keep a single whitespace as a prefix. It then removes the prefix from the dataframe column and, if the boolean value is true, adds a single whitespace as a prefix. Finally, it returns the modified dataframe.

- AI detectorの例文
  - 基礎編4.Openai事例⑪. Python to natural languageの出力（右の囲い）を判定
- 判定例
  - ② crossplag
  - ③ copyleaks

AI生成



AI生成



5. Stable Diffusion
①. Stable Diffusion概要

- Stable Diffusionとは
  - テキストから自動で画像を生成するAIです。
  - https://stablediffusionweb.com/
  - openaiのDALL·E2と競合、オープンソースのためapiをサーバにインストールすれば無料で使用できます。但し、サーバは、高性能gpuを搭載したものが必要になります。
  - インストールサービスなどを提供
- 事例
  - ① tet2image
    - 「a professional photograph of an astronaut riding a horse」のテキストから画像生成

- サーバスペック
Rtx3080, i9, 32GB
Ubuntu18.04
なお、web版はokだが通常のpythonは動作しない。メモリ不足：12GB（GPU)必要？