Azure x LangChain　実行結果報告

2024/4/5　岡田　隆之

LangChainをAzureのOpenAI APIを用いて実行してみた。

実行結果は、現時点で以下の表のようになった。

|  |  |  |  |
| --- | --- | --- | --- |
| 項目 | 内容 | 実行出来たか | 項目の重要性 |
| LLM応答 | 単純なチャット応答 | 出来た | ◎ |
| RAG | Cognitive Searchに登録し、登録情報の参照を実行。参照方法として、キーワード検索、ベクトル検索、LLMの意味検索がある。 | 出来た。キーワード検索、ベクトル検索、LLMの意味検索、そしてそれらの組み合わせ検索もできた。 | ◎ |
| on your data | GUIの操作がほとんどで、自動化には合わない | 未（やらない予定） | X |
| LangChain | RAGで重要な存在。自社データ、Web情報、外部API活用など、LLMが自動プランニングし、一回の応答で何回もLLMを実行し組み合わせて応答 | 未。使用料制限で、Azure OpenAI Serviceのグレードアップが必要と言われた | ◎ |
| Semantic Kernel | Microsoft社版の簡易LangChain。たくさんのことはできないが、シンプルなのが良いという。しかし、Web検索の追加の仕方とか不明事項多 | 出来た。ネットからSummarizeやCodingスキルのファイルセットをダウンロードし、実行させることが出来た。 | △ |
| Function Calling | LangChainの関数実行しない版。LLMが1回しか回らないため、複雑な動きが出来ない。 | 未。必要性を感じれば実行する | X |
| Prompt Flow | GUI。 | 未。自動化できないため合わない | X |
| プラグイン | Data analysisなどの高度なプラグインはまねできないのでぜひ検討したい。 | 未⇒LangChainかSemantic Kernelから呼び出せるということ。 | 〇（ここまでいらないかも） |
| Fine Tuning | 独自のLLMを作る。とても高くなりそう | 未。高そうなため了解を得ないと。 | X |

◆実行時のメモ

AzureのLLM実行には奇妙な点がある。設定項目の無駄だ。次のような設定項目がある：

・Azure OpenAI

APIキー、エンドポイントURL、謎の”azure”宣言、APIバージョン、作成したデプロイ名

・Azure OpenAI Embedding

APIキー、エンドポイントURL、作成したデプロイ名、デプロイしたモデルのタイプ

・Cognitive Search

APIキー、エンドポイントURL（Cognitive Searchのもの）、情報グループ名（インデックス名）

下の画像のように、例えば、一番初めのベースのLLM応答で上記のAzure OpenAIの設定項目を行っている。azure宣言。これは、openaiライブラリのときは必要。langchainでAzureChatOpenAIを使うときにも求められる！（⇒省略したい！！）APIキーは必須。エンドポイントもこれは許容できる（APIキーがわかればエンドポイント、あるいはエンドポイントがわかればAPIキーがわからないのかな？？）。APIバージョンも求められる！（APIに紐づいてないの！？困る。余計な要求に思う）また、デプロイ名も必要。これは自分が動くようにリソースを切っていないといけず、どれを使うかを指定しないといけないのでいる。



このように、Azureとの連携はUI/UXに納得いかない部分が多々ある。

メンテナンスも大変になるため無駄な記述が改善されていくといいなと思う。

終わり■