

大規模言語モデル、自然言語処理による アプリケーション改修予測サービス

笠 春樹 haruki@toki.waseda.jp

システム開発における問題点

システム設計の修正を繰り返してシステム品質の向上を進めているが、初期段階で、システムの改修が必要と思われる箇所を特定できていれば、期間と改修コストを短縮できる可能性がある。

手法・ツールの適用による解決

LLMに過去に改修した大量の改修履歴、ソースコードなどをインプットし、設計内容と改修内容とソースコードの相関から、改修すべき箇所や考慮すべき箇所などをガイドし、実装の不備などをチェックする機能を提供する。

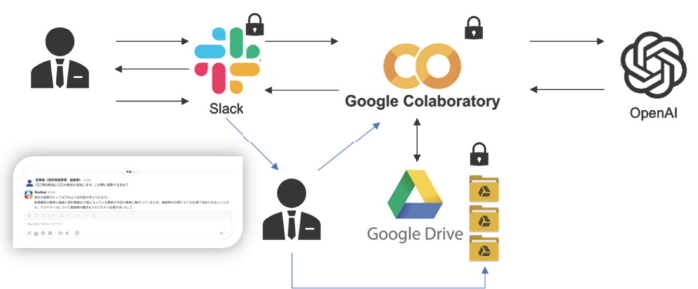
システム概要

AIナレッジを踏まえた開発プロセスとすることで、
(1)システム実装のミス防止(2)効率的でスピーディーな開発(3)IT人材不足の解消 を実現



システム構成図

- ・利用者はSlack上でやり取りのみで利用可能
- ・RAG登録もクラウドに格納するだけで実現する
- ・LLMの業務利用にはセキュリティ面の考慮が必要



仮説検証のアプローチ

| Step | 内容 |
|------|--|
| 0 | サンプルとなるデータの作成 (データクリーニング、項目選定、JSON変換) |
| 1 | OpenAIのPlaygroundを利用したRAG利用 |
| 2 | Assistants & Retrievalを使ってColaboratory上で実行 |
| 3 | Embeddingモデルを利用してモデルの精度向上 |
| 4 | Slackを活用したチャットの構築 |
| 5 | チューニング・利活用の考察 |

システム改善の勘所

| # | チューニングの検討 | 内容 |
|---|------------|--|
| 1 | モデルのVerアップ | ・LLMのモデルのバージョンアップ ・変更(Amazon Kendora, Watson) |
| 2 | RAGの調整 | 有用な情報に絞って登録する。 |
| 3 | プロンプトの調整 | 固定プロンプトが正しい結果を導くような文章にチューニングする。 |
| 4 | 項目の選定 | より良い結果を生み出す項目を取捨選択する。 |
| 5 | 追加データの改善 | 追加データは、モデルの項目を意識した状態で登録する。 |
| 6 | 適用範囲の変更 | PoCの結果にとらわれず、利用価値を生み出す利用方法を探求する。 |

< LLM出力サンプル >

この情報に基づき、ソフトウェア開発チームは以下の修正案を提出することができます:
 ・UI/UXの改善: 「検索ボタン」をより目立つ場所に配置し、エラーメッセージを分かりやすく表示すること、ユーザーとの関わりをより深くし、操作性を向上させます。
 ・APIの最適化: 既存のAPIエンドポイントに対して、非同期処理を導入し、応答時間を短縮させることを検討することをお勧めします。
 ・セキュリティ強化: ユーザー入力データを適切に検証し、不正なデータがシステムに入力されることを防ぐことが重要です。
 これらは提案に過ぎず、詳細な検証と、開発チームでの検討が必要です。なお、このファイルは100%完成品ではありません。完全な情報提供を目的としたものではありません。