

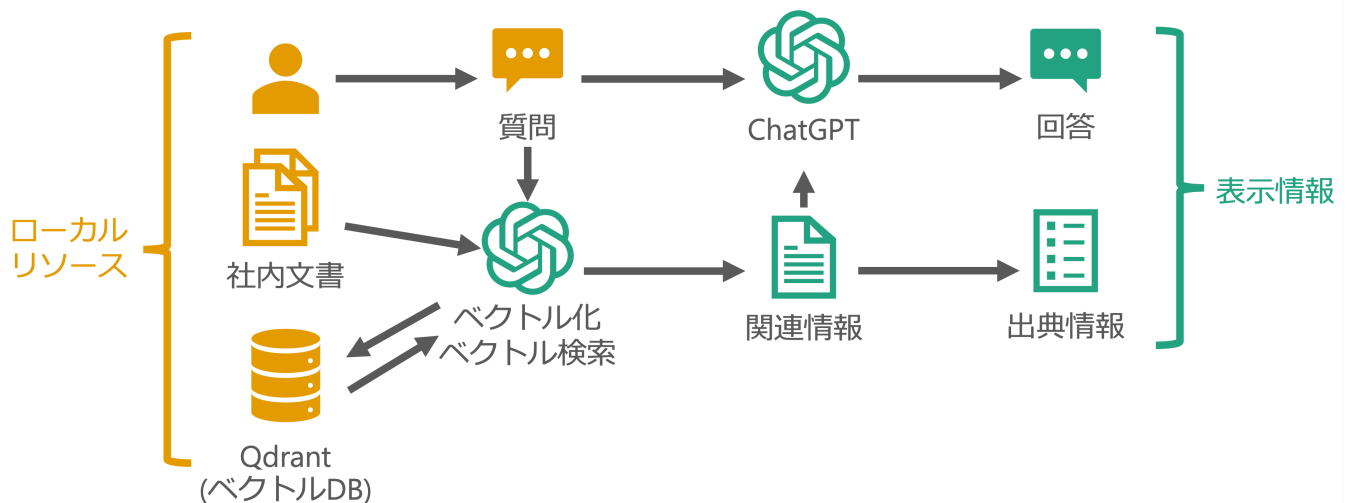
KadaGPT アプリ用マニュアル

概要

本アプリは、chatGPTを始めとするLLMや自然言語処理を用いて組織内文書を検索、質問に対する応答を生成することで、オフィスワークを支援することを目的に開発中です。現在基本的な質問応答処理、必要な関連情報の部分選択処理を実装しています。

構成図

動作概要



アプリは文書登録タスクと質問タスクに分けられます。それぞれ以下のようなフローになります。

文書登録

1. 文書をOpenAIの提供するモデルを使ってEmbedding(ベクトル化)する。
2. Embeddingした文書をベクトルDBに保存する。

質問

1. 質問をOpenAIの提供するモデルを使ってベクトル化する。
2. ベクトル情報を使って、ベクトルDB内を検索し、関連情報をとってくる。
3. 質問と関連情報を組み込み、引用情報を提示するよう指示したプロンプトをChatGPTに投げる。
4. 回答を取得し表示する。
5. 回答より、使用したpdfファイルを引用として表示する。

サーバ構成

dockerコンテナとしてバックエンド側ではFastAPIコンテナ、Qdrantコンテナ、DBコンテナ、Nginxコンテナが、フロントエンド側ではNext.jsコンテナおよびKeycloakコンテナが動作しています。FastAPIへの直接アクセスには、サーバの8080ポートにアクセスします。フロントエンドを用いた対話では、ブラウザから<http://FQDN>にアクセスします。

技術要件

必要マシンスペック

- CPU: スペック不問
- dGPU: 必要無し
- RAM: 8G以上
- ディスク: 10GB以上 登録したドキュメントに応じて増加

必要ソフトウェア

- OS: Dockerが動けばOK
- コンテナツール: Docker

その他

- インターネットに接続できること
 - dockerでイメージをビルドするため
 - OpenAI APIに問い合わせるため
 - api.openai.comに接続する

使い方

グループ新規登録方法

グループの追加では、API側のみ操作します。

1. <http://192.168.100.103:8080/docs>にアクセスします。
2. `/chat/groups`のGETメソッドタブを展開し、右上のTry it outを押下した後、Executeを実行します。
3. 現状のグループ一覧が表示されるので、確認します。
4. 新規追加を行うので、`/chat/groups`のPOSTメソッドタブを展開します
5. Try it outを押し、Request body内にあるjsonのname欄を希望するグループ名に編集します
6. Executeを押し、Responsesに200番でnameとidが返ってきていることを確認します
7. 登録作業が終わったら、`/chat/groups`のGETメソッドからちゃんと登録されているか確認します

ユーザ新規登録方法

ユーザを新規で登録する場合、現状ではフロント側とAPI側の両方に登録作業が必要です。ユーザ登録に必要な情報は以下です。

- ユーザ名: 英数字
- メールアドレス
- 初期パスワード
- 氏名
- 所属するグループID

API側

1. <http://192.168.100.103:8080/docs>にアクセスします。

2. `/chat/users`のGETメソッドタブを展開し、右上のTry it outを押下した後、Executeを実行します。
3. 現状のユーザー一覧が表示されるので、すでに登録されていないか確認します
4. 新規登録を行うので、`/chat/users`のPOSTメソッドタブを展開します
5. Try it outを押し、Request body内のjsonを編集します。
 - name: ユーザの名前です。アルファベットを利用してください
 - email: ユーザのemailです。
 - password: 現状API側では利用していないので、そのまま大丈夫です。
 - group_id: ユーザが所属するグループを指定します。グループ一覧は`/chat/groups`のGETメソッドタブから確認してください。
6. 必要事項を入力したのち、Executeを押下します。
7. Responsesに200番で登録情報+ユーザIDが返却されていることを確認します
8. 登録作業が終わったら、`/chat/users`のGETメソッドからちゃんと登録されているか確認します

フロント側(Keycloak)

1. `http://2x5u16fb5yk4fr.aa0.netvolante.jp/keycloak` もしくは `http://192.168.100.103/keycloak` にアクセスします。
2. 管理者権限でのログインを求められるので、ユーザ名admin, パスワードhogehogeでログインします。
3. 左上のレルム選択タブからKadaGPTを選択します。
4. 左のメニューからUsersを選択します。
5. ユーザー一覧が表示されるので、Add userからユーザ追加画面に移動します。
6. 必要項目を埋めていきます
 - Required user actionsはユーザが次回ログインしたときのアクションを指定します。一時的にパスワードを発行し、あとから変更させたい場合は、Update Passwordを選択します。
 - Email verifiedは、emailが検証済みかどうかを切り替えます。emailでログインしたい場合はYesとしておきます。
 - Select a locale: 日本語のままで良いかと思えます
 - Username: ユーザ名を指定します。ここは、API側で指定したnameと必ず一致させてください。
 - Email: ユーザのEmailです。ここもAPI側と一致させることを推奨します。
 - First name: ユーザの名を入力します。アプリの挙動には影響しないので、空欄でも構いませんが、その場合は初回ログイン時にユーザ自身が直接埋めるよう要求されます。
 - Last name: ユーザの姓を入力します。First nameと同様です。
 - Groups: ここでのグループはアプリの挙動と関係ないので、編集しなくて大丈夫です。
7. 項目を埋め終わったら、Createより作成します。
8. Credentialsタブより、初期パスワードの設定ができます。
9. Set passwordをクリックします。
10. 設定したいパスワードを2回入力します。
- Temporaryは、次回パスワードをリセットするかどうかを決定するものです、管理者側で一時的に付与し、ユーザ自身でパスワードを変更してもらいたい場合はここをYesのままにしておきます。
11. Saveを押し、パスワードの登録を完了します。
12. ユーザー一覧より、ユーザが登録されていることを確認します。

ユーザ情報更新

ユーザ情報を変更する場合も、フロント側とAPI側の両方に登録作業が必要です。ユーザ登録に必要な情報は以下です。

- ユーザ名: 英数字
- 所属するグループID
- その他, emailなど

API側

1. <http://192.168.100.103:8080/docs>にアクセスします。
2. </chat/users>のGETメソッドタブを展開し、右上のTry it outを押下した後、Executeを実行します。
3. 現状のユーザー一覧が表示されるので、変更したいユーザを確認します
4. 変更を行うので、</chat/users>のPATCHメソッドタブを展開します
5. Try it outを押し、[user_id](#)およびRequest body内のjsonを編集します。
 - [user_id](#): 変更対象のユーザIDです。
 - [name](#): ユーザの名前です。アルファベットを利用してください
 - [email](#): ユーザのemailです。
 - [password](#): 現状API側では利用していないので、そのまま大丈夫です。
 - [group_id](#): ユーザが所属するグループを指定します。グループ一覧は</chat/groups>のGETメソッドタブから確認してください。
6. 必要事項を入力したのち、Executeを押下します。
7. Responsesに200番でユーザ情報が返却されていることを確認します
8. 登録作業が終わったら、</chat/users>のGETメソッドから登録されているか確認します

フロント側(Keycloak)

1. <http://2x5u16fb5yk4fr.aa0.netvolante.jp/keycloak> もしくは <http://192.168.100.103/keycloak> にアクセスします。
2. 管理者権限でのログインを求められるので、ユーザ名admin, パスワードhogehogeでログインします。
3. 左上のレルム選択タブからKadaGPTを選択します。
4. 左のメニューからUsersを選択します。
5. ユーザー一覧が表示されるので、編集したいユーザをクリックします。
6. 情報の変更を行います。
7. Saveを押し、パスワードの登録を完了します。
8. ユーザー一覧より、ユーザが登録されていることを確認します。

コレクション(知識ベース)登録

KadaGPTは、PDFから取得した情報をコレクションに蓄積します。グループとコレクションは1対多で紐づけられます。また、現状のコレクションは1つのグループにしか所属できないため、PDF情報を共有したい場合は他のグループのコレクションに追加で登録する必要があるので注意してください。

1. <http://192.168.100.103:8080/docs>にアクセスします。
2. /chat/collections/{group_id}のGETメソッドタブからグループ内の既存コレクションを確認します。

- 3. Try it outを押し、対象のグループIDをgroup_idに入力したのち、Executeを押すことで確認できます。

コレクションを新規作成する場合

- 1. /chat/collectionsのPOSTメソッドより登録を行います。
- 2. Request bodyには、コレクション名、所属させるグループID、作成したユーザIDを指定します。
- 3. Executeし、200番台のレスポンスコードと共にコレクションのIDが返却されることを確認します。

コレクションにPDFを登録する場合

- 1. /chat/collections/{collection_id}のPOSTメソッドより登録を行います。
- 2. collection_idには登録対象のコレクションIDを、create_user_idにはPDFを登録するユーザIDを入力します。
- 3. Request bodyのfilesより、アップロードしたいファイルをローカルより選択します。形式はPDFのみです。複数ファイルを選択することができます。
- 4. Executeし、アップロードされたことを確認します。PDFファイル毎にIDが振られるはずです。
- 5. /chat/collections/{collection_id}/documentsのGETメソッドより、コレクションに登録されているPDFを確認します。

なお、PDFを更新する場合は、一度/chat/collection/{collection_id}/{document_id}のDELETEメソッドより当該のドキュメントを削除した後、もう一度登録手順を実施してください。

再起動時の操作

再起動時には、コンテナの起動を確認します。~/kada-gpt/front-endおよび~/kada-gpt/back-endでdocker compose ps -aで起動状態を確認します。正常に起動していれば、~/kada-gpt/front-endでは以下のようにになっているはずです。

NAME	IMAGE	COMMAND	SERVICE
CREATED	STATUS	PORTS	
app	front-end-app	"docker-entrypoint.s..."	app 7
days ago	Up 7 days		
auth	quay.io/keycloak/keycloak:24.0	"/opt/keycloak/bin/k..."	auth 7
days ago	Up 7 days	8080/tcp, 8443/tcp	
auth-db	mariadb:10.11.6-jammy	"docker-entrypoint.s..."	auth-db 7
days ago	Up 7 days	3306/tcp	
traefik	traefik:2.11.2	"/entrypoint.sh trae..."	traefik 7
days ago	Up 7 days	0.0.0.0:80->80/tcp, :::80->80/tcp	

また、~/kada-gpt/back-endでは以下のようになっています。

NAME	IMAGE	COMMAND	SERVICE
CREATED	STATUS	PORTS	
api	back-end-api	"poetry run gunicorn..."	api
8 days ago	Up 8 days		
back-end-db-1	mariadb:10.11.6-jammy	"docker-entrypoint.s..."	db
8 days ago	Up 8 days	3306/tcp	

```
back-end-vector-store-1  qdrant/qdrant:v1.6.1  "./entrypoint.sh"  vector-  
store  8 days ago  Up 8 days  6333-6334/tcp  
proxy  nginx:1.25.3-alpine  "/docker-entrypoint...."  proxy  
8 days ago  Up 8 days  0.0.0.0:8080->80/tcp, :::8080->80/tcp
```

上記のようになっていれば、自動でサービスを利用できます。もし起動していない場合は、`~/kada-gpt/front-end`および`~/kada-gpt/back-end`で`docker compose up -d`とコマンドを入力してください。

ブラウザ経由のチャットを行う

1. <http://2x5u16fb5yk4fr.aa0.netvolante.jp>にブラウザからアクセスします。
2. 右上に**LOGIN**と表示されているので、予め与えられたユーザIDとパスワードでログインします。
3. 左のメニューからスレッドを作成します。
4. スレッド作成時には、スレッド名とChatGPTのモデル、関連文書取得数、知識ベース(コレクション)選択、手法の選択を行います。
5. スレッド作成後は、ChatGPTがコレクションに登録した知識を使って応答する対話を利用できます。

サポート情報

もし不具合などが発生しましたら、現地あるいはSSH経由にて対処させていただきます。
連絡先はこちら

- 増田嶺(香川大学) s24g357@kagawa-u.ac.jp
- 岩本和真(香川大学) s24g351@kagawa-u.ac.jp
- 道信祐成(香川大学) s24g359@kagawa-u.ac.jp

いつでもメンテナンスできるよう、SSH用のポートを外部へ開けていただけると幸いです。