Guide Utilisateur – PDF *Extractor*

Attijariwafa Bank

Résumé

Ce guide utilisateur présente les instructions détaillées pour exploiter l'application PDF *Extractor,* développé par Attijariwafa Bank.

Introduction

Bienvenue dans le guide utilisateur de PDF Extractor développée par Attijariwafa Bank, cette application offre une solution professionnelle pour extraire des données structurées à partir de documents PDF et les convertir en fichiers CSV. Que vous ayez à traiter des formulaires administratifs, des contrats ou tout autre document textuel, PDF Extractor met à votre disposition une interface intuitive et des technologies avancées, notamment l'intelligence artificielle, pour optimiser vos processus. Ce guide vous accompagnera de manière détaillée dans l'installation, la configuration et l'utilisation de l'application, garantissant une expérience fluide et efficace adaptée à vos besoins professionnels.

1 Prérequis

Avant d'utiliser PDF *Extractor, assurez-vous de disposer des éléments suivants* :

-Un système d'exploitation Windows 11.

-Python 3.8 ou une version supérieure

installée.

-Les dépendances Python nécessaires, installées via un fichier requirements.txt (Voir section Installation).

-Le serveur Ollama en cours d'exécution avec le modèle mistral:7b-instruct chargé.

2 Installation

Pour installer et configurer PDF Extractor de manière optimale, veuillez suivre les étapes ci-dessous :

1. Téléchargez le répertoire du projet PDF Extractor depuis le dépôt officiel fourni par Ilyas Larabi et Attijariwafa Bank.
2. Ouvrez une invite de commande ou un terminal et naviguez vers le répertoire du projet à l'aide de la commande appropriée.
3. Créez un environnement virtuel pour isoler les dépendances en exécutant la commande "python -m venv venv".
4. Activez l'environnement virtuel afin de travailler dans un espace dédié : sous Windows, exécutez "venv\Scripts\activate".
5. Installez les dépendances nécessaires en exécutant la commande "pip install -r requirements.txt" dans l'invite de commande activée.

   fastapi
   pydantic
   pandas
   pymupdf4llm
   langchain_community
   requests
   uvicorn

6. Installez Ollama depuis le site officiel puis démarrez le serveur Ollama, requis pour le traitement par intelligence artificielle, en exécutant "ollama serve" dans une seconde invite de commande.
7. Chargez le modèle "mistral:7b-instruct" en utilisant la commande "ollama pull mistral:7b-instruct" pour assurer la compatibilité avec l'application.

3 Utilisation

Pour exploiter pleinement les fonctionnalités de PDF Extractor, suivez ces étapes détaillées afin d'extraire des données à partir de vos fichiers PDF et de les convertir en fichiers CSV :

1. Vérifiez que le serveur Ollama est en cours d'exécution sur votre système. Ce serveur est essentiel pour le traitement des données via le modèle mistral:7b-instruct.

2. Lancez l'application en exécutant la commande "python main.app.py" depuis le répertoire du projet dans une invite de commande ou un terminal activé avec l'environnement virtuel.

3. Une fenêtre de dialogue s'affichera automatiquement, vous permettant de sélectionner un fichier PDF. Naviguez dans votre système de fichiers, choisissez le fichier souhaité, puis cliquez sur "Ouvrir".

4. L'application procèdera à l'extraction des données du PDF sélectionné et les convertira en un fichier CSV nommé "output.csv". Ce processus peut inclure une analyse initiale par expressions régulières, suivie d'un traitement par intelligence artificielle si nécessaire.

5. Une fois le traitement terminé, consultez le fichier "output.csv" généré dans le répertoire du projet. Ce fichier contiendra les données extraites sous une forme structurée, prête à être utilisée dans d'autres applications.

4 Explication de l'application

1. **Sélection du Fichier PDF** : Le processus commence lorsque vous lancez l'application avec la commande "python main.app.py". Une interface graphique, alimentée par Tkinter, s'ouvre pour vous permettre de sélectionner un fichier PDF depuis votre système Windows 11. Une fois le fichier choisi, son chemin d'accès est capturé pour le traitement ultérieur.

2. **Conversion en Format Markdown** : L'application utilise la bibliothèque PyMuPDF4LLM pour ouvrir le PDF sélectionné et convertir son contenu textuel en format markdown. Cette étape préserve la structure du document (comme les en-têtes ou les listes) et prépare les données pour une analyse plus approfondie, en ignorant les parties non pertinentes comme les préambules si nécessaire.

3. **Extraction Initiale avec Expressions Régulières** : Les données markdown générées sont ensuite analysées à l'aide d'expressions régulières définies dans le module extractor.py. Cette étape identifie et extrait des champs spécifiques (par exemple, nom, email, fonction) en se basant sur des motifs prédéfinis, avec un nettoyage des valeurs pour éliminer les bruits ou espaces superflus.

4. **Traitement Avancé avec Intelligence Artificielle** : Pour les champs non détectés ou complexes, l'application fait appel au modèle mistral:7b-instruct d'Ollama via LangChain. Un prompt structuré guide le modèle pour extraire les données manquantes en respectant un schéma Pydantic, validant ainsi les informations extraites (par exemple, signatures ou commentaires).

5. **Validation des Données** : Les données extraites, qu'elles proviennent des expressions régulières ou de l'IA, sont validées à l'aide de schémas Pydantic définis dans user_data.py. Cette étape garantit que toutes les informations respectent la structure attendue, remplissant les champs absents par une valeur par défaut comme "N/A".

6. **Conversion en Fichier CSV** : Les données validées sont transmises à une API FastAPI hébergée localement. Cette API, définie dans main.py, convertit les données en un DataFrame pandas, qui est ensuite sauvegardé sous forme de fichier CSV nommé "output.csv" dans le répertoire du projet, prêt à être utilisé.

7. **Retour à l'Utilisateur** : Une fois le processus terminé, l'application affiche un message de confirmation dans la console et vous invite à consulter le fichier "output.csv". Toute erreur éventuelle (par exemple, un serveur Ollama non démarré) est également signalée pour faciliter le dépannage.

5 Dépannage

Cette section vous guide à travers les solutions aux problèmes courants que vous pourriez rencontrer lors de l'utilisation de PDF Extractor, développé par Ilyas Larabi et Attijariwafa Bank. Suivez ces recommandations pour résoudre les éventuels dysfonctionnements :

- **Erreur : Serveur non démarré** - Si vous recevez un message indiquant que le serveur Ollama n'est pas accessible, vérifiez qu'il est en cours d'exécution. Ouvrez une invite de commande séparée, exécutez "ollama serve", et assurez-vous que le modèle mistral:7b-instruct est chargé avec "ollama pull mistral:7b-instruct". Redémarrez l'application après avoir corrigé cela.

- **Fichier non sélectionné** - Si la fenêtre de dialogue Tkinter s'ouvre mais que aucun fichier n'est traité, assurez-vous d'avoir sélectionné un fichier PDF valide en cliquant sur "Ouvrir" après avoir navigué jusqu'au fichier. Vérifiez également que l'extension du fichier est ".pdf".

- **Erreur d'extraction** - Si l'application échoue à extraire des données, cela peut indiquer que le PDF contient des éléments non textuels (comme des images scannées) ou un format non pris en charge. Assurez-vous que le document contient du texte exploitable et essayez avec un autre fichier PDF. Consultez les logs dans la console pour des détails supplémentaires.

- **Performance lente ou plantage** - En cas de traitement lent ou d'arrêt inattendu, vérifiez que votre système dispose de ressources suffisantes (mémoire et processeur) pour exécuter Python, Ollama, et les dépendances. Fermez les applications inutiles et redémarrez l'environnement virtuel avant de relancer "python main.app.py".

- **Fichier CSV non généré** - Si "output.csv" n'apparaît pas, confirmez que l'API FastAPI est en cours d'exécution (consultez la console pour le message de démarrage sur http://127.0.0.1:8000). Assurez-vous également qu'aucune erreur réseau n'interrompt la communication entre l'interface graphique et l'API.

Pour tout problème persistant, consultez les messages d'erreur affichés dans la console, qui fournissent des indices sur la cause.

6 Conclusion

PDF Extractor, développé par Ilyas Larabi en partenariat avec Attijariwafa Bank, représente une solution robuste et efficace pour automatiser l'extraction de données à partir de documents PDF et leur conversion en fichiers CSV. Grâce à ses fonctionnalités avancées, telles que la sélection dynamique de fichiers, l'extraction structurée via des expressions régulières et l'intelligence artificielle, ainsi que la validation rigoureuse des données, cette application répond aux besoins des utilisateurs cherchant à optimiser leurs processus de gestion documentaire. En suivant les étapes d'installation, d'utilisation et les conseils de dépannage fournis dans ce guide, vous pouvez exploiter pleinement ses capacités pour améliorer votre productivité.

Nous vous encourageons à explorer toutes les possibilités offertes par PDF Extractor et à tirer parti de son potentiel pour vos projets professionnels. Pour toute question supplémentaire, assistance technique ou suggestions d'amélioration, n’hésitez pas à contacter l’équipe de support d’Ilyas Larabi ou d’Attijariwafa Bank. Ce guide vise à vous accompagner tout au long de votre expérience avec cette application innovante.