

Unstructured 0.13.7

| | | |
|--|--|---------------------|
| U N S T R U C T U R E D | Outil d'ingestion de données pour LLM | |
| | Système d'exploitation : | Multiplateforme |
| | Développé par : | Unstructured |
| Open source, Apache-2.0 | Personne de contact : | katy.fokou@smals.be |

Fonctionnalités

[Unstructured](#) est un outil modulaire pour la construction de pipelines d'ingestion de données non-structurées dans le but de construire des applications utilisant les grands modèles de langage typiquement de type RAG (questions-réponses). L'outil d'ingestion est capable de traiter des données sous différents formats provenant de sources diverses. Il se présente sous forme d'API ou plateforme *low-code* en version commerciale et sous forme d'un package python en version open-source.

Les fonctionnalités clés d'Unstructured sont les suivantes :

- Extraction d'éléments structurés tels que les titres, tables, images de documents tels que les PDF, les fichiers Powerpoint, Word, Excel, XML ou les pages Web (html).
- Division du texte en blocs (*chunking*) avec différentes méthodes, taille fixe, par titre, par fenêtre glissante, ...
- Génération d'*embeddings* ou représentations vectorielles de blocs de texte par intégration de modèles de tierce-partis.
- Enrichissement des données avec des résumés de tables et d'images (non disponible en open-source).

How does it work?

To get your data RAG-ready, the Unstructured Platform moves it through the following process:

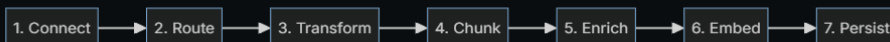


Fig1: Pipeline de traitement de données

Conclusions & Recommandations

Unstructured est un outil open-source incontournable pour le traitement de données pour leur utilisation dans des applications basées sur les LLM (*large language model*). La plateforme est très versatile et de nombreuses fonctionnalités intéressantes sont disponibles dans la version gratuite.

Tests & Résultats

Pour évaluer efficacement les capacités d'Unstructured nous avons installé la version open-source de l'outil et testé les fonctionnalités citées précédemment à l'exception de la génération de résumés d'images et de tables.

```
from IPython.core.display import display, HTML
display(HTML(tables[1].metadata.text_as_html))
```

```
/tmp/ipykernel_455318/1760248645.py:1: DeprecationWarning: Importing display from IPython.core.display is deprecated.
from IPython.core.display import display, HTML
```

| Metrics | AlfWorld | | ScienceWorld | | HotPotQA | | FEVER | |
|-----------|----------|--------|--------------|--------|----------|--------|-------|--------|
| | SR(%) | EX(\$) | AR | EX(\$) | SR(%) | EX(\$) | SR(%) | EX(\$) |
| Z-CoT | | | | | 0.01 | 0.95 | 0.39 | 1.07 |
| F-CoT | 0.43 | 98.60 | 16.58 | 272.22 | 0.32 | 5.73 | 0.61 | 2.25 |
| CoT-SC | 0.57 | 105.37 | 15.24 | 274.33 | 0.33 | 7.86 | 0.62 | 321 |
| SayCan | 0.60 | 113.61 | 12.36 | 125.71 | | | | |
| ReAct | 0.57 | 152.18 | 15.05 | 356.03 | 0.34 | 66.00 | 0.63 | 22.20 |
| Reflexion | 0.71 | 220.17 | 1939 | 72448 | 0.39 | 112.49 | 0.68 | 26 |

Fig1. Table extraite d'un document PDF

Le premier test consiste en la validation de la fonction « partition » pour l'extraction de textes de documents et la fonction de *chunking*. La fonction « partition » peut être utilisée en spécifiant le format du document au préalable. Les documents sous format PDF, html, msg ont été testés et correctement traités par l'outil. Pour les documents PDF en particulier, plusieurs réglages sont possibles dont la gestion de l'OCR, l'utilisation d'un modèle pour une extraction précise, etc. On notera en particulier l'extraction sans-faute

d'images et de tables (fig1) de documents PDF ainsi que la détection de titres et liens hypertexte de page html. Cependant, la détection de titre dans les PDF n'est pas très fiable, beaucoup de morceaux de textes non pertinents sont identifiés comme titres. De même, la fonction de chunking par titre n'atteint pas les performances attendues. En plus des fonctions spécifiques de partition, Unstructured propose une fonction générique capable de détecter automatiquement le format du document, la langue du document et d'appliquer le moteur d'extraction approprié. Les multiples tests appliqués à des documents de différents formats et de différentes langues sont tous positifs.

La deuxième série de tests porte sur l'établissement d'un pipeline complet de traitement de documents (fig2). Le pipeline mis en place contient les éléments suivants : connexion à la source, extraction, chunking, *embedding* et stockage dans une base de données vectorielle (Chroma). Le pipeline peut s'exécuter dans un script python ou en ligne de commande. Unstructured propose par défaut de multiples connecteurs pour les sources de données brutes et les destinations de données traitées. Le pipeline est facile à mettre en œuvre et le traitement des données se fait rapidement.

```
if __name__ == "__main__":
    Pipeline.from_configs(
        context=ProcessorConfig(work_dir=False, num_processes=4),
        indexer_config=LocalIndexerConfig(input_path=LOCAL_FILE_INPUT_DIR),
        downloader_config=LocalDownloaderConfig(),
        source_connection_config=LocalConnectionConfig(),
        partitioner_config=PartitionerConfig(
            partition_by_api=False,
            strategy="hi_res",
            hi_res_model_name='detectron2_mask_rcnn',
            additional_partition_args={
                "split_pdf_page": True,
                "split_pdf_allow_failed": True,
                "split_pdf_concurrency_level": 15
            }
        ),
        chunker_config=ChunkerConfig(chunking_strategy="by_title", chunk_max_characters=1000),
        embedder_config=EmbedderConfig(embedding_provider="langchain-huggingface",
                                       embedding_model_name="intfloat/multilingual-e5-large"),
        destination_connection_config=ChromaConnectionConfig(
            access_config=ChromaAccessConfig(headers={"Authorization": "Basic()"}),
            path=CHROMA_DATA_PATH,
            collection_name=CHROMA_COLLECTION,
        ),
        stager_config=ChromaUploadStagerConfig(),
        uploader_config=ChromaUploaderConfig()
    ).run()
```

Fig2. Ex. de configuration d'un pipeline

Conditions d'utilisation & Budget

Unstructured est disponible comme bibliothèque Python open-source et sous licence Apache ou, sous forme d'API pour un prix allant de 1 à 10\$ par 1000 pages.