Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Impact de l'extraction de texte pour l'identification de la structure d'un document depuis des documents PDF du domaine financier

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Mei GAN

Directeur de mémoire :

Cyril Grouin

Encadrant :

Juyeon Kang

Année universitaire 2020/2021

Remerciements

Je voudrais tout d'abord adresser toute ma reconnaissance à mon encadrant de mémoire M. Cyril GROUIN, pour sa patience, sa disponibilité et surtout ses conseils, qui ont contribué à alimenter ma réflexion.

Je désire aussi remercier ma tutrice de stage Mme. Juyeon KANG, chef de l'équipe Data Science chez Fortia, pour sa confiance, son temps et ses conseils tout au long de mon projet.

Mes vifs remerciements vont également à toute l'équipe pédagogique de pluriTAL pour leur attention, leur patience et leur aide permanente tout au long du master.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

Je tiens à remercier mes collègues au sein du pôle Data Science de Fortia pour leur accueil sympathique et leur soutien permanent. Un grand merci à Sandra BELLATO pour son aide précieuse à la relecture et à la correction de mon mémoire.

Je tiens à témoigner toute ma gratitude à Liyun YAN et Nanzhong HE pour leur confiance et leur soutien inestimable.

Mes parents, pour leur soutien constant et leurs encouragements.

RÉSUMÉ

Le domaine de l'extraction d'informations automatisée de texte à partir de documents PDF est d'une grande importance car le format PDF est toujours l'un des plus populaires pour la représentation et l'échange d'informations, en particulier dans le monde de la finance. En outre, il est également important pour un système d'extraction d'informations à partir de documents financiers de s'assurer de la fiabilité des données extraites. Ce mémoire a donc pour objectif de comparer plusieurs outils de reconnaissance de texte afin d'identifier la structure des documents PDF financiers. Nous présenterons les différents types de documents PDF et les méthodes utilisées pour l'extraction de texte et de détection de la structure. Ensuite, nous nous attarderons sur les évaluations de l'extraction de texte, de la détection de format et de la détection de structure de documents PDF.

Mots-clés : PDF, outils d'extraction PDF, extraction de texte, détection de structure de document, évaluation des performances

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION

Présentation générale

De nos jours, de nombreux documents électroniques sont sauvegardés sous forme de documents PDF (Portable Document Format) qui est un langage de description de page présenté par la société Adobe System en 1992. Cette technologie en langage Postscript permet de représenter sous forme de vecteurs tous les éléments composant une page (texte, polices d'écriture, graphiques, images, etc.) dans un fichier unique. Dans le domaine de la finance, les rapports d'investissement tels que les documents d'Offering Memorandum, les Prospectus et les KIID (Key Investor Information Document) qui fournissent les informations nécessaires aux investissements s'appliquant aux fonds sont souvent en format PDF et ils permettent aux entreprises de gestion de fonds d'échanger avec les investisseurs sur toutes les plate-formes ou applications spécifiques tout en conservant la mise en page d'origine.

Néanmoins, la recherche d'une solution lisible de tous les systèmes d'exploitation implique des difficultés pour identifier et extraire le texte contenu dans ces documents PDF et détecter la structure dans ces fichiers. En effet, la méthode d'encapsulation empêche de traiter ces documents PDF comme nous traiterions un simple fichier texte. D'autres part, les informations pertinentes dans un document financier peuvent être présentées sous plusieurs formes, avec des mises en page et structures différentes selon chaque société de gestion. En conséquence, un visualiseur PDF prend simplement les coordonnées x et y de chaque caractère et dessine l'ensemble dans une page que l'utilisateur peut visualiser. Cette méthode n'est pas facilement exploitable pour l'extraction d'informations.

La conversion de documents PDF en texte et la détection du format et de la structure sont des étapes fondamentales pour ces systèmes d'extraction d'informations. La précision et la qualité de ces étapes impactent directement les étapes suivantes telles que la préparation des données pour l'entraînement des modèles en apprentissage automatique et la performance de ces modèles.

Ce mémoire est le résultat d'un projet de recherche en traitement automatique des langues effectué dans une entreprise spécialisée en intelligence artificielle appliquée au domaine de la finance, Fortia Financial Solution au sein de l'équipe Data Science et dans le cadre du master Traitement Automatique des Langues (TAL) de l'Inalco. L'objectif du projet est de comparer et d'évaluer la performance de différents outils d'extraction de contenu, de détection de format et de détection de la structure de fichiers PDF.

Problématique et Objectif

Bien que les documents PDF présentent des avantages de diffusion sous différentes plateformes, la plupart des documents convertis dans ce format ne sont pas structurés. Cette particularité pose problème lors de la conversion en texte et la dé-

tection des informations sur les polices et la structure. Un bloc de texte issu d'une conversion peut ainsi ne pas correspondre à un bloc de texte physique ou sémantique. Vraisemblablement, deux raisons sont à l'origine du problème : l'ordre de création du fichier et la génération faite par des outils automatiques[Rigamonti et al., 2004]. Pour simplifier, la position des objets PDF (texte, images, graphiques, styles, etc.) dans un fichier correspond à leur ordre de génération. La génération des fichiers PDF faite par des convertisseurs automatiques pourrait être une autre justification à l'ordre chaotique des objets. En l'état actuel, un fichier PDF se limite à décrire l'apparence d'un document et toute notion structurelle a été abandonnée.

Pour un système d'extraction d'informations basé sur les documents PDF, il est très important d'avoir un outil qui permet d'extraire les composants basiques (texte, tableaux, images et graphiques) et de détecter également les formats enrichis et la structure d'un fichier PDF[Lovegrove and Brailsford, 1995]. Ces informations de fichier PDF sont essentielles pour un système d'extraction d'informations.

L'objectif de ce mémoire est d'offrir, entre autres, une réflexion autour des questions suivantes :

— Quels types d'erreurs surviennent lors de la conversion de PDF en texte ?
— Quels formats un outil peut-il détecter, et quels formats ne peut-il pas détecter ?
— Existe-t-il un outil capable de détecter la structure des documents PDF, et si oui, comment organise-t-il la représentation de cette structure ?

En basant notre travail de recherche sur la comparaison de différents outils internes et externes, nous espérons fournir des éléments de réponse à ces questions générales.

Organisation du mémoire

Nous présenterons dans un premier temps une description du projet DOC Reader dans lequel s'inscrit notre travail de recherche. Dans un deuxième temps, nous exposerons les différents types de documents PDF ainsi que les approches et outils existants pour l'extraction de texte et la détection de structure. Le troisième chapitre sera consacré à la présentation et la constitution de jeux de données pour les évaluations. Par la suite, nous nous arrêterons sur les expériences que nous avons menées et analyserons les résultats obtenus. Enfin, nous nous attarderons sur la conclusion de notre étude, où nous parlerons des limites et des perspectives de travaux futurs.

Première partie

Contexte général

CHAPITRE 1

CONTEXTE INDUSTRIEL

Sommaire

1.1 Introduction

La problématique de recherche abordée dans ce mémoire repose sur l'évaluation des performances d'outils d'extraction de données PDF sous plusieurs aspects. Nous présenterons d'abord le projet DOC Reader qui est une application d'extraction d'informations qui propose aux clients un moyen d'analyse automatique de documents financiers. Nous décrirons ensuite les besoins et contraintes spécifiques à ce projet d'extraction d'informations.

1.2 Présentation du projet OM Reader

DOC Reader (Figure 1.1) est une application d'extraction d'informations qui propose aux sociétés financières telles que les entreprises d'investissement et les banques un moyen d'entraîner des modèles avec leurs documents PDF et d'extraire automatiquement des informations de documents financiers, notamment ceux décrivant les fonds d'investissement des acteurs financiers et accessibles au public via leur site. À la suite de l'extraction d'informations d'un document, une liste d'informations clés extraites est proposée selon les besoins des clients. L'application est surtout orientée vers les banques, les autres services financiers qui gèrent des inventaires de leurs produits et les data analystes qui analysent des données afin de pouvoir proposer une stratégie d'investissement.

Cette application comprend principalement deux fonctionnalités (Figure 1.2) : 1) l'entraînement d'un modèle d'extraction d'informations personnalisé avec les documents PDF de clients ; 2) la prédiction automatique d'informations sur des documents PDF avec un modèle personnalisé ou un modèle pré-entraîné par Fortia.

1) La fonctionnalité d'entraînement des modèles d'extraction d'informations repose sur quatre étapes :

Création d'un corpus de documents PDF : Il s'agit de créer un corpus et télécharger des documents PDF, par exemple des documents financiers comme des

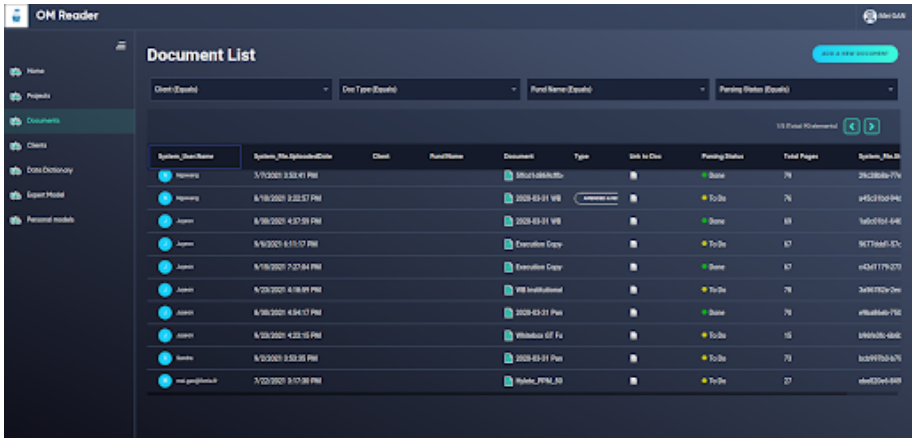


FIGURE 1.1 – Interface de l'application DOC Reader

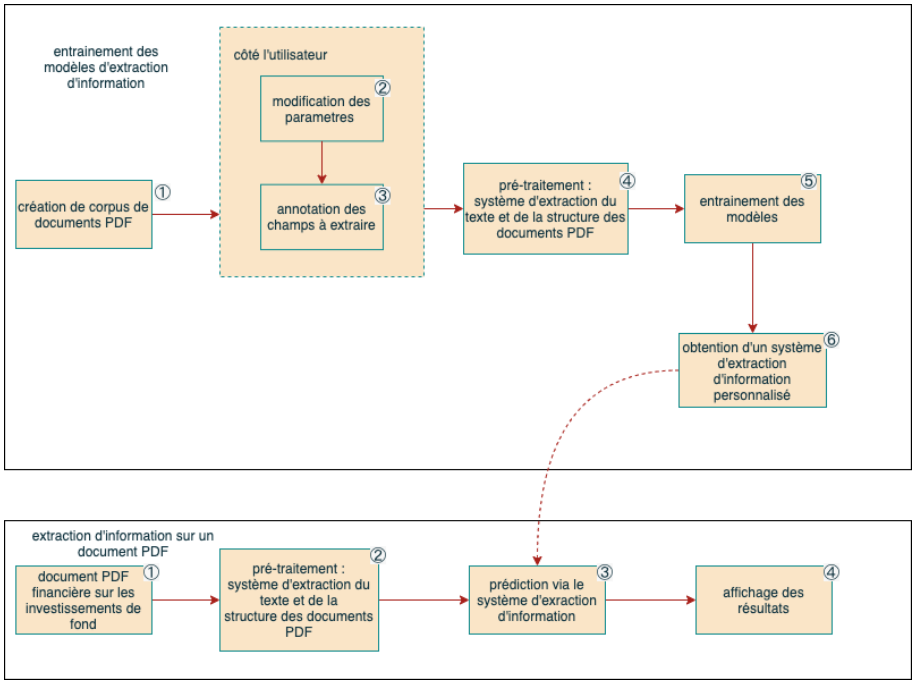


FIGURE 1.2 – Schéma du entraînement du modèle et de prédiction sur un document PDF

Offering Memorandum (documents financiers sur les fonds d'investissement) dans l'application ;

Modification des paramètres et annotation des champs : Ensuite, les utilisateurs peuvent choisir et paramétrer pour chacun des champs qu'ils souhaitent extraire, une des méthodes d'extraction proposée par Fortia. Nous proposons (Figure 1.3) au moins trois méthodes d'extraction d'informations pour les champs qui n'ont pas besoin d'être annotés et quatre méthodes pour les champs qui ont besoin d'être annotés. Des champs (l'exemple dans l'annexe A.1) sur des informations importantes du domaine financier, surtout sur les investissements des fonds sont proposés par l'application. Les utilisateurs peuvent par exemple préciser des sections spécifiques dans les documents où extraire des informations pour un champ en ajoutant des mots clés ou des titres dans le paramétrage du champ. Après avoir défini les méthodes d'extraction des champs, les utilisateurs vont annoter des données si besoin sur notre outil appelé Annotation Tool qui permet aux clients de créer des étiquettes pour chaque champ et d'annoter les documents du corpus sur l'application.

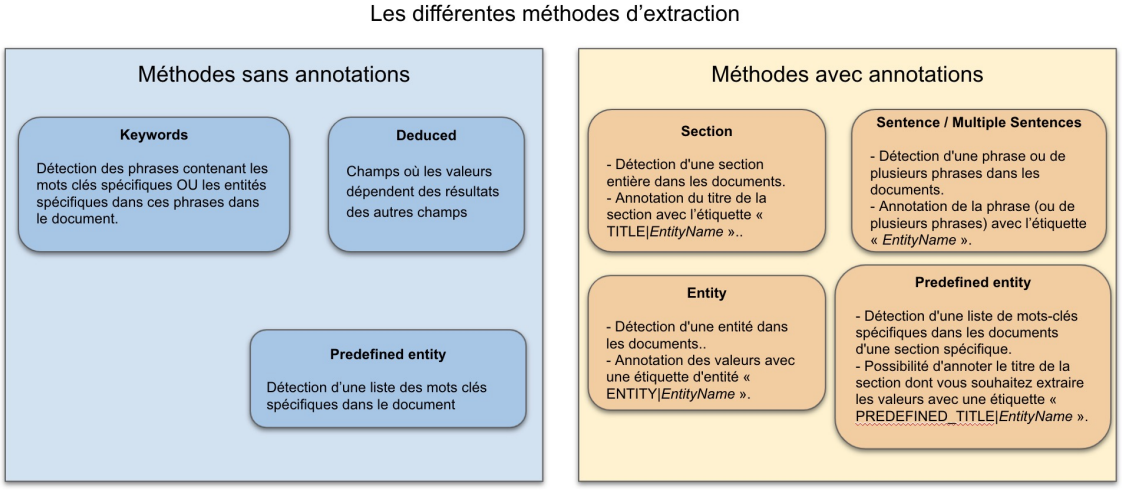


FIGURE 1.3 – Schéma récapitulatif de méthode d'extraction pour personnaliser les méthodes d'extraction

Pré-traitement : Il s'agit de pré-traiter les documents avec un système d'extraction de texte et de détection de structure de documents PDF. Ce système propose un processus en deux étapes afin de récupérer le texte et les coordonnées puis de détecter la structure des documents PDF :

- La récupération du texte et des coordonnées est réalisée par un outil appelé pdfExtract, intégrant un logiciel tier, Vintasoft Net SDK[1], qui convertit les documents PDF en texte et récupère des informations basiques de caractères telles que les polices et les coordonnées.
- L'outil docParser (un outil propriétaire de Fortia) permet de détecter la structure de documents PDF à partir des résultats de l'étape précédente. Il effectue un processus d'identification de titres en deux étapes :
  - La reconstitution de l'ordre de lecture dans un bloc de texte se fait à partir des coordonnées de texte. Chaque sous-chaîne est définie par ses coordonnées x et y (le repère O (0,0) est le coin en haut à gauche du document), la longueur, la hauteur et les informations concernant l'espace entre les

1. www.vintasoft.com

caractères et les mots. L'algorithme de docParser est développé pour différents documents financiers notamment comme les Prospectus, KIID et par la suite, nous avons ajouté des analyses heuristiques pour certains types de documents comme les Offering Memorandum. Ces informations suffisent donc à recréer les lignes et les mots. Ensuite, il filtre des textes tels que les notes de bas de page ou les pieds de page qui sont des informations inutiles pour le système d'extraction d'informations.

— L'identification des titres repose sur l'identification des polices, de la mise en forme (gras, italique, taille) ou de l'existence de symboles typographiques au début de chaîne.

Ces informations seront utilisées pour trouver les titres ou blocs de texte correspondant aux paramètres choisis par les utilisateurs.

Entraînement des modèles : Si les utilisateurs souhaitent extraire d'autres champs que ceux proposés par l'application, ou s'ils veulent un modèle spécifique pour leurs propres documents, nous avons la possibilité d'entraîner un modèle personnalisé avec leurs données. Nous avons un modèle de reconnaissance d'entités nommées pour les champs annotés marqués *ENTITY* et un modèle d'extraction d'informations contextualisé pour les champs dont les méthodes d'extraction sont « sentence » ou « multiple sentences ».

Nous proposons également un modèle pré-entraîné si les clients n'envisagent pas d'entraîner leur propre modèle.

2) La fonctionnalité d'extraction d'informations sur un document PDF :

Pré-traitement : Cette étape est identique à l'étape de pré-traitement précédente qui fait appel au système d'extraction de texte et à la détection de structure de document.

Prédiction via le système d'extraction d'information : Des algorithmes d'extraction d'informations basés majoritairement sur l'intelligence artificielle sont appliqués. Comme présenté précédemment, nous avons un modèle de reconnaissance d'entités nommées pour les champs annotés marqués *ENTITY* qui a été entraîné sur des données générales. De plus, le modèle d'extraction d'informations contextualisé a été entraîné avec 149 documents Offering Memorandum que nous avons annotés sur 43 champs proposés dans le cadre d'un projet X (anonyme).

Affichage des résultats : L'application fournit la liste des informations détectées par les modèles et un lien vers une interface de visualisation (l'Annotation Tool)(Figure 1.4) où les résultats sont surlignés directement dans le document PDF original avec également la liste des informations détectées à droite.

1.3 Problématique

Lors de l'analyse des erreurs de prédiction des modèles, nous avons remarqué que certains problèmes venaient de l'étape de la conversion de PDF en texte. Cela impacte fortement même les résultats prédits par les modèles dans la mesure où le système d'extraction de texte et de détection de la structure constitue une étape initiale pour entraîner des modèles d'apprentissage automatique. La qualité des textes et des structures extraites impactent directement la performance des modèles d'ex-

1. www.vintasoft.com

FIGURE 1.4 – Visualisation des résultats sur l'Annotation Tool

traction d'informations. Il est donc indispensable d'évaluer ces étapes pour avoir une image plus complète sur les limites des outils.

Le système d'extraction de texte pour les documents de type Offering Memorandum est assez complexe, et repose sur plusieurs outils de traitement de documents. Un outil appelé pdfExtract est utilisé pour la récupération de texte et des coordonnées et l'outil docParser est utilisé pour la détection de structure à partir des sorties de pdfExtract. Ainsi, plusieurs approches ont été adoptées et développées afin d'évaluer chacune de ces étapes.

1.4 Conclusion

Nous avons présenté le contexte général dans lequel notre stage s'est déroulé et ce mémoire a été rédigé. À partir de la présentation du projet DOC Reader, nous avons pu formuler une problématique concernant l'extraction d'informations textuelles qui résume les objectifs de ce mémoire : Quelle est la performance du système d'extraction de texte pour extraire les informations nécessaires ?

Dans le chapitre 2, nous effectuerons un état de l'art des différents types de fichiers PDF et des outils d'extraction de texte de documents PDF, et qui nous permettra d'enrichir cette problématique et de la transformer en problématique de recherche dans le chapitre 4.

CHAPITRE 2

ÉTAT DE L'ART

Sommaire

2.1 Introduction

A ce jour, la norme dominante pour archiver les documents électroniques est le PDF, créé à l'origine en tant que format propriétaire par Adobe Systems Incorporated au début des années 1990 [Warnock, 1991] et qui est devenue une norme ISO ouverte (qui a été officiellement adoptée en 2008 et adopté par Adobe via une licence publique qui accorde une utilisation sans redevance)[Orion, 2007].

Cette section est composée de deux sous-sections, qui présentent respectivement les différents types de documents PDF et l'état actuel du point de vue des outils d'extraction de texte sous différents aspects : 1) l'état de l'art en ce qui concerne la détection de texte et de format de caractères ; 2) l'état de l'art concernant la détection des structures de document PDF.

2.2 Types de PDF existants

Les documents PDF peuvent être créés directement ou indirectement en convertissant le contenu. Lors de la création directe de PDF, toutes les capacités du composant peuvent être utilisées. La conversion de contenu au format PDF est le moyen le plus simple et le plus courant d'obtenir des fichiers PDF mais généralement les fichiers obtenus ne tirent pas de modèle d'imagerie Adobe (Adobe Imaging Model) de haut niveau en décrivant la sortie à un niveau très bas[van der Knijff, 2009a]. En revanche, une fois le PDF créé indirectement, toutes les informations de haut niveau sont perdues et ne peuvent plus être récupérées. Cela rend les tâches dans le cadre

de l'extraction et du traitement des données beaucoup plus difficiles, car la plupart des informations doivent être récupérées sur la base des informations incomplètes incluses dans le document[van der Knijff, 2009a].

Les documents PDF peuvent être classés en trois types différents, selon la façon dont le fichier a été créé. La manière dont le fichier a été créé définit aussi si le contenu du fichier PDF (texte, images, tableaux) peut être accessible ou s'il est « scellé » dans l'image de la page.

2.2.1 PDF natif

Un document PDF typique contient des milliers d'objets de bas niveau, de multiples mécanismes de compression, différents formats de police, des lignes, des courbes, des vecteurs et du contenu auxiliaire[Whitington, 2011]. Il est souvent créé directement à partir des logiciels de traitement de texte tels que Microsoft Word. Un PDF natif est en quelque sorte une norme composite, unifiant au moins trois technologies de base[Berg et al., 2012] :

1. Un sous-ensemble du langage de « programmation » de la page PostScript, qui supprime des constructions telles que des boucles et des branches, mais inclut toutes les opérations graphiques pour dessiner des éléments de mise en page, du texte et des images.
2. Un système d'intégration de polices qui permet à un document de « transporter » une grande variété de polices (dans divers formats), comme cela peut être nécessaire pour garantir l'affichage tel que le document a été créé.
3. Un système de stockage structuré, qui organise divers objets de données, par exemple des images et des polices, à l'intérieur d'un document PDF.

Tous les objets de données dans un fichier PDF sont représentés de manière visuelle, sous la forme d'une séquence d'opérateurs qui, lorsqu'ils sont interprétés par un moteur de rendu PDF, dessinent le document sur un canevas de page. En conséquence, dans ces documents (Figure 2.1), les caractères de texte et les méta-informations ont un caractère numérique correspondant. Des logiciels de visualisation tels d'Adobe nous permettent d'effectuer des recherches facilement, de sélectionner, de copier ou même écrire directement sur le document.

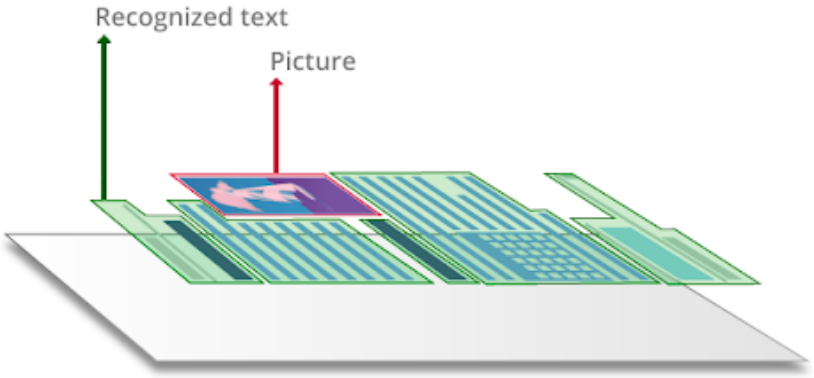


FIGURE 2.1 – PDF natif

Source de la figure : https ://pdf.abbyy.com/learning-center/pdf-types/

2.2.2 PDF scannés

Un PDF scanné (Figure 2.2) est un document qui contient seulement les images photographiées ou numérisées de pages, sans une couche de texte. Ces documents peuvent avoir été à l'origine des fichiers PDF natifs, mais en effectuant une numérisation, ils ne deviennent plus qu'une image avec l'avantage que l'intégrité du fichier peut être maintenue. La modification de PDF scannés est très difficile et nécessite des connaissances techniques ainsi que des outils appropriés. Travailler avec des documents simplement constitués d'images peut être difficile car la couche de texte manquante empêche également l'utilisateur d'effectuer des actions de recherche et de copie[I. Y. Korneev et al., 2015].

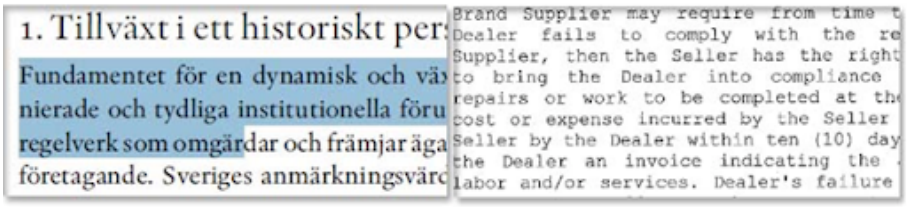


FIGURE 2.2 – Différence entre PDF natif et PDF scanné

Les documents PDF peuvent être exploitables grâce à l'application OCR (Optical Character Recognition) avec laquelle un calque texte est ajouté, normalement sous l'image de la page. Cela rend le document éditable et sélectionnable.

2.2.3 PDF multicouches

Grâce au processus de reconnaissance de texte, les caractères et la structure du document PDF sont analysés. Un PDF multicouches (Figure 2.3) contient une couche de texte qui est placée au-dessous de la couche image. Ce type de document est sélectionnable et éditable comme les documents PDF natifs.

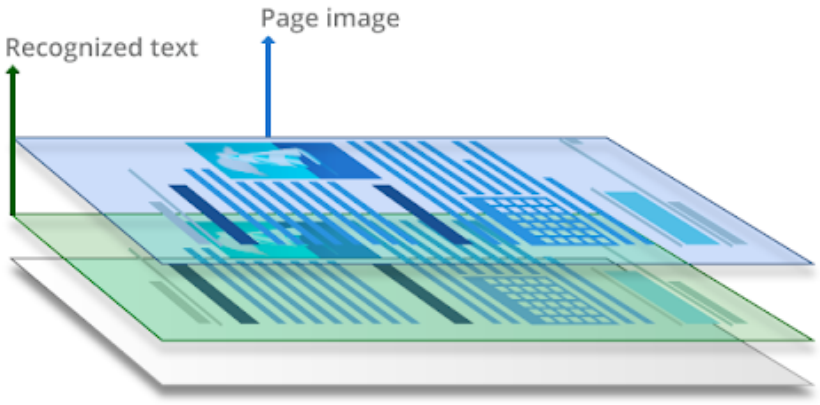


FIGURE 2.3 – PDF multicouches

Source de la figure : https ://pdf.abbyy.com/learning-center/pdf-types/

Si la qualité des textes n'est pas satisfaisante, la technologie de reconnaissance de texte OCR peut être appliquée pour convertir le document au format souhaité.

2.3 Extraction de texte de document PDF

[Bui et al., 2016a] définissent l'extraction de données comme un processus standard dans le développement de revue systématiques. Étant donné que de tels processus reposent généralement sur des actions manuelles qui sont lentes, coûteuses et sensibles aux erreurs, les systèmes automatisés d'extraction de texte et de traitements ultérieurs sont très demandés. Un autre article de Bui et al.[Bui et al., 2016b] déclare que les systèmes d'extraction d'informations (IE) ont un énorme potentiel pour aider les humains à effectuer des tâches d'extraction de texte longues. L'inconvénient est que la plupart des systèmes d'extraction d'informations ne sont pas conçus pour l'extraction de données à partir de documents PDF. La raison pour laquelle la plupart des outils IE ne se concentrent pas sur les documents PDF, même si les documents PDF sont une source d'extraction très courante et importante, en particulier dans le domaine de la finance, est la grande complexité liée à la mise en œuvre de tels outils [Bui et al., 2016b]. Les PDF confondent le contenu narratif avec les métadonnées ainsi qu'avec le texte semi-structuré qui est très difficile à différencier pour les algorithmes automatisés. Selon [Damerow et al., 2017], la conversion d'images et de fichiers PDF en texte brut est en demande constante, car les étapes de traitement ultérieures telles que la reconnaissance d'entités nommées ou la modélisation de sujets reposent sur le contenu textuel. Le format de fichier PDF représente toujours la majorité des rapports financiers même si d'autres formats de données tels que les pages web HTML ont également été utilisés. Après avoir extrait des textes à partir de documents PDF, d'autres processus tels que le pré-traitement de textes comme tokenization, segmentation, nettoyage, etc, l'analyse de structure de documents, l'extraction d'informations peuvent être appliqués pour enrichir et extraire le contenu textuel.

2.3.1 Extraction de texte et de format de caractère de document PDF

[Bast and Korzen, 2017] décrivent le processus d'identification des mots dans les documents PDF comme non trivial. Ils mentionnent l'espacement comme un défi principal car les espaces blancs peuvent varier considérablement entre les lettres et les mots et peuvent également différer d'une ligne à l'autre, par exemple si le texte est justifié. Par conséquent, il est difficile de définir des mots en ne considérant que les distances d'espacement. D'autres défis peuvent être les sauts de ligne dans des mots et des caractères spéciaux qui ne peuvent pas être identifiés. Différents styles de PDF (colonne unique, colonnes multiples) entraînent des problèmes concernant l'identification correcte de l'ordre des mots. Il est également difficile de déterminer les limites des paragraphes car ces limites peuvent être interrompues par des figures, des tableaux et des sauts de page.

De nombreux outils qui sont capables d'extraire automatiquement les textes à partir de documents PDF sont déjà disponibles pour le public mais une grande variation en qualité et en nombre de fonctionnalités (comme l'identification des limites de paragraphe, l'identification de l'ordre de lecture ou l'identification de structure) rendent difficile l'évaluation de ces approches. En 2017, [Bast and Korzen, 2017] ont mené une analyse approfondie de plusieurs approches et ont donné un aperçu détaillé des performances et de la qualité des résultats ainsi que des fonctionnalités fournies par chaque approche. Bast et Krozen ont analysé et évalué 14 outils en fonction de dix attributs au cours du processus d'extraction.

Cette évaluation montre qu'en ce qui concerne les critères les plus importants comme la reconnaissance de mot, tous les algorithmes donnent des résultats similaires. En ce qui concerne la détection automatique des limites de paragraphe, l'outil pdfact montre une meilleure performance parmi les outils testés.

De plus, la détection du format des caractères de document PDF tels que la police, la taille et la couleur des mots, est également complexe. De préférence, toutes les polices utilisées dans une mise en page sont également incluses dans le fichier PDF lui-même. Cela garantit que le fichier peut être affiché et imprimé tel qu'il a été créé par le convertisseur. Il existe deux mécanismes pour inclure des polices dans un PDF [1] :

1. Embedding - une copie complète de l'ensemble des glyphes de caractères d'une police est stockée dans le PDF. Le programme de police d'origine peut être intégré dans le fichier PDF. PDF décrit un ensemble de 14 polices standard qui peuvent être utilisées sans définition préalable[van der Knijff, 2009b].
2. Subsetting - seuls les caractères réellement utilisés dans la mise en page sont stockés dans le PDF. Par exemple, si le caractère « € » n'apparaît pas dans le texte, ce caractère n'est pas inclus dans la police. Cela signifie que les fichiers PDF avec des polices subsetting sont moins volumineux que les fichiers PDF avec des polices avec embedding. Pour les polices en subsetting, le nom de la police est précédé de 6 caractères aléatoires et d'un signe plus.

Lorsqu'une police est intégrée dans un PDF, toutes les données de police ne sont pas incluses. D'ailleurs, d'autres informations, telles que les données sur les ligatures, ne sont pas pertinentes dans le PDF, de sorte que ces données ne sont pas incluses. De plus, les programmes de polices sont soumis à des droits d'auteur, et le propriétaire du droit d'auteur peut imposer des conditions dans lesquelles un programme de polices peut être utilisé[van der Knijff, 2009b]. Par conséquent, vu que la plupart des polices sont protégées par le droit d'auteur, l'utilisation d'un extracteur pour récupérer les formats de caractères est illégal.

Des outils qui sont capables d'extraire les formats des caractères :

— **pdfminer** [2] est un outil qui extrait les coordonnées des textes, les noms de police, les tailles de police, le sens d'écriture (horizontal ou vertical) pour chaque segment de texte en plusieurs formats possibles (text, html, xml). Néanmoins, il ne reconnaît pas le texte dans les images.

— **pdfact** [3] est un outil intégré dans un système de gestion des documents de recherche (Icecite [4]), qui est capable d'extraire les textes et les rôles sémantiques (titre, auteur, année de publication, référence, etc.) à partir des articles scientifiques. En ce qui concerne l'extraction de texte, il est basé sur une approche basée sur des règles qui analyse les distances, les positions et les polices de caractères, de mots et de lignes de texte afin d'extraire les paragraphes du texte.

— **FineReader PDF** [5] d'Abbyy est un logiciel commercial qui permet de l'extraction des informations dans des documents PDF. Le site officiel d'Abbyy offre une version d'essai d'extraction pour la conversion en différents formats (txt,

1. https ://www.prepressure.com/pdf/basics/fonts
2. https ://github.com/euske/pdfminer
3. https ://github.com/ad-freiburg/pdfact
4. https ://icecite.cs.uni-freiburg.de/
5. https ://www.abbyy.com/

html, word, odt, etc.). Le format HTML sortie de l'outil contient les informations sur les formats de caractères PDF : la police, les tailles de police et les règles typographiques. Mais ces fichiers HTML ne contiennent pas d'informations sur les couleurs des textes.

— **pdfplumber**[6] est une librairie Python qui est capable d'extraire des informations détaillées sur les caractères des textes et également les tableaux. Les fichiers sorties sont des fichiers csv ou json.

— **Vintasoft**[7] est un produit commercial qui offre des services pour lire, écrire et modifier et extraire les fichiers PDF sur WinForms, WPF et ASP.NET. Il fournit des informations sur le format des caractères ainsi que leurs coordonnées.

Dans le but d'effectuer l'extraction de texte à partir de documents numérisés ou dactylographiés, une autre technologie appelée reconnaissance optique de caractères (OCR) est nécessaire pour pré-traiter les documents. Parmi ces outils, seuls FineReader PDF et pdfact ont cette fonctionnalité.

2.3.2 Reconnaissance optique de caractères

[Bennamoun et al., 2001] définissent la reconnaissance optique de caractères comme « le processus de conversion d'une représentation d'image matricielle d'un document dans un format qu'un ordinateur peut traiter ». Selon [Islam et al., 2016], l'OCR (Optical Character Recognition) est définie comme « le processus de numérisation d'une image de document en ses caractères constitutifs ». Ce domaine couvre plusieurs sous-catégories de l'informatique comme le traitement des images et le traitement automatique des langues, la reconnaissance des formes, l'intelligence artificielle et les systèmes de bases de données[Bennamoun et al., 2001].

Pour convertir les documents PDF scannés en fichiers PDF exploitables, un logiciel OCR est utilisé pour reconnaître et extraire les données des pages du document dont le contenu visuel des fichiers originaux sous forme d'images. [Masum et al., 2018] décrivent l'OCR comme une technologie très importante dans le contexte de l'analyse automatisée de documents commerciaux. Il permet d'archiver des copies numériques de documents qui ne peuvent pas être manipulés. Cela permet à l'extraction automatisée des connaissances et de rendre plus efficace le processus de travail dans l'entreprise en effectuant des opérations de recherche basées sur des mots clés spécifiques à un domaine.

Comme mentionné précédemment, la technologie OCR permet la conversion de fichiers des images en données exploitables par machine qui peuvent être éditées, recherchées, stockées numériquement et utilisées dans des processus automatisés. Par conséquent, les institutions et les entreprises utilisent couramment la technologie OCR pour numériser leurs documents. Selon[Patel et al., 2012], la précision de l'OCR dépend fortement des algorithmes de pré-traitement et de segmentation du texte. Différents styles, tailles, et couleur de fond ou image de fond complexes rendent difficile la détection de lettres dans les images. Les caractères avec peu de différence visible (par exemple, le chiffre 0 et la lettre O) sont très difficiles à distinguer les uns des autres et le texte intégré dans un fond très sombre peut difficilement être reconnu par l'OCR. Le premier système commercial d'OCR pour la numérisation de documents bureautiques a été publié en 1955. De nos jours, de nombreux outils d'OCR de qualité

6. https ://github.com/jsvine/pdfplumber
7. https ://www.vintasoft.com/

variable concernant leur précision de reconnaissance sont disponibles sur le marché, mais peu d'entre eux sont libres d'utilisation et open source[Patel et al., 2012].

Selon [Chaudhuri et al., 2017], les systèmes d'OCR reposent sur plusieurs techniques différentes telles que : le balayage optique, la segmentation de localisation, le prétraitement, la segmentation, la représentation, l'extraction de caractéristiques, l'apprentissage et la reconnaissance et le post-traitement.

1. Le **balayage optique** convertit les images à plusieurs niveaux en images noir et blanc à deux niveaux. Pour ce faire, un seuil est défini où les niveaux de gris au-dessus de ce seuil sont classés comme blancs et les niveaux de gris inférieurs sont classés comme noirs.
2. La **segmentation de localisation** est essentielle pour localiser les régions textuelles et les séparer des régions constituées d'images avant de poursuivre le processus de reconnaissance. Le plus souvent, chaque caractère est isolé et reconnu individuellement sans son contexte. Des problèmes techniques de segmentation surviennent lorsque des caractères touchent leurs voisins ou lorsque des graphiques sont confondus avec du texte et vice versa. Un autre défi est d'éviter le bruit qui est dans les documents après qu'ils aient été scannés sous forme d'images.
3. Le **prétraitement** vise à produire des données simples à analyser pour les systèmes OCR. Le prétraitement se décompose en trois composants : réduction du bruit, normalisation des données et compression des informations à retenir.
4. La **segmentation** vise à diviser les caractères en ses sous-composants. Il existe plusieurs approches différentes pour ce problème mais surtout la segmentation de l'écriture cursive reste à résoudre.
5. La manière dont les images sont transmises au module de reconnaissance est gérée dans l'étape de **représentation**. Cela vise à maximiser la variabilité entre les classes en extrayant les informations les plus représentatives des données.
6. L'**extraction de caractéristiques** est l'un des problèmes les plus difficiles de la reconnaissance des formes. Les caractéristiques essentielles des symboles sont capturées pour pouvoir les classer par la suite. Les techniques fréquemment utilisées pour l'extraction de caractéristiques sont la correspondance de patron, la distribution de points et l'analyse structurelle. La deuxième partie de cette étape se concentre sur la classification afin d'identifier chaque caractère et de l'affecter à la classe correspondante.
7. Les principales approches utilisées pour l'étape d'**apprentissage et de reconnaissance** sont la correspondance de patron, les techniques statistiques, les techniques structurelles et les réseaux de neurones artificiels (ANN). Les approches peuvent être utilisées seules ou en tant que variantes combinées puisqu'elles ne s'excluent pas les unes les autres.
8. Le **post-traitement** vise à regrouper, détecter les erreurs survenues dans les étapes précédentes et les corriger. Étant donné qu'aucun système de reconnaissance ne peut garantir des résultats complètement corrects, la détection d'erreurs en utilisant le contexte est utilisée pour améliorer le résultat. En plus, l'utilisation de dictionnaires est une solution populaire pour détecter et corriger efficacement les erreurs. Les mots qui ne sont pas dans le dictionnaire sont classés comme des erreurs et corrigés au mot existant le plus similaire. D'une

part ce processus est très fiable et efficace mais d'autre part cette tâche prend beaucoup de temps.

2.4 Détection de structure de document PDF

La conservation des rôles sémantiques des documents PDF, tels que les titres et les sections, est une tâche importante pour permettre un traitement des documents PDF plus approfondi. Selon [Rigamonti et al., 2004], un des aspects principaux de l'analyse de la structure d'un document est l'extraction des propriétés physiques et logiques des régions d'un document. Les propriétés physiques mettent en évidence la topologie du document, tandis que les propriétés logiques représentent la fonction des régions (e.g. titre, section, figure, etc.). Bien que le domaine de l'analyse de document et la détection de la structure soit en permanente évolution, tous les buts n'ont pas été atteints. Par exemple, dans des documents multi-colonnes, l'ordre d'apparition des blocs de texte ne reflète en général pas l'ordre de lecture. La difficulté de définir l'ordre du texte et des paragraphes est difficile à deux niveaux :

1. Parfois, il n'y a pas de bonne réponse. Alors que conventionnellement, les documents avec une composition d'une seule colonne ont un ordre de lecture naturel, les documents avec des mises en page plus complexes (e.g. deux colonnes ou mixtes à une ou deux colonnes) ont un ordre de lecture plus difficile à identifier.

FIGURE 2.4 – Exemple d'une page avec un bloc de texte inséré

Il arrive que des portions de phrase ou des mots isolés n'apparaissent pas dans leur contexte mais de manière isolée. Comme dans l'exemple (Figure 2.4), il n'est pas clair si l'encart doit apparaître avant, après ou parmi l'article à côté duquel il est placé.

Certaines imperfections pourraient déprendre de l'historique du document et des logiciels qui ont servi à le produire.

2. Deuxièmement, même lorsque la réponse est claire pour un humain, déterminer un ordre de paragraphe robuste est un problème très difficile à résoudre pour l'IA. Il existe des cas où l'ordre correct des paragraphes ne peut être décidé qu'en comprenant le contenu du texte.

CONFIDENTIAL OFFERING MEMORANDUM

Dated: February 25, 2021 *Continuous Offering*

Capitalized terms used but not defined in this cover page shall have the meaning specified in the Glossary of Terms.

THE ISSUER:

Name:	Dynamic Alpha Performance Fund
Head Office:	Dynamic Funds Tower 1 Adelaide Street East, 28th Floor Toronto, Ontario, M5C 2V9
Phone Number	1-800-268-8186
Email Address:	invest@dynamic.ca
Fax Number:	1-800-361-4768
Currently Listed or Quoted:	**These securities do not trade on any exchange or market.**
Reporting Issuer:	No
SEDAR Filer:	No

THE OFFERING:

Securities Offered:	An unlimited number of multiple series of trust units are being offered hereby on a continuous basis. See "Investing in the Fund". Each Unit within a particular series will be of equal value; however, the value of a Unit in one series may differ from the value of a Unit in another series. Each series shall have the attributes and characteristics as set out under the heading "Units of the Fund".
Price Per Security:	The purchase price of a Unit is an amount equal to its Net Asset Value per Unit. The Net Asset Value per Unit for subscription orders which are received and accepted by the Manager prior to 4:00 p.m. (Toronto time) on a Valuation Day will be calculated as of that Valuation Day. The Net Asset Value per Unit for subscription orders received and accepted on or after 4:00 p.m. (Toronto time) on a Valuation Day will be calculated on the next Valuation Day. See "Portfolio Valuation and Net Asset Value".
Minimum/Maximum Offering:	**There is no minimum or maximum offering. You may be the only purchaser.**

FIGURE 2.5 – Exemple d'une page contenant deux colonnes

(Figure 2.5) présente un exemple où l'ordre de sens de cette page est d'abord un ordre naturel pour le titre de section et premier paragraphe. Ensuite, l'ordre de lecture de deux colonnes est de gauche à droite en prenant le contexte du texte. En revanche, un outil pourrait extraire la colonne gauche entièrement vu que certains titres sont très attachés.

Par conséquent, l'analyse des structures contenues dans le fichier PDF se révèle indispensable afin de segmenter une composition complexe en sous-zones textuelles, d'image et graphiques. Il n'existe pas beaucoup d'outils qui nous permettent de détecter la structure des documents PDF. Des recherches [Doucet et al., 2013], [Teufel, 2010] et [McConnaughey et al., 2017] ont été menées pour extraire la structure des documents principalement à partir d'articles et de livres scientifiques. [Guedes and Silva, 2021] ont proposé une approche d'apprentissage supervisé entraîné sur un grand ensemble de données pour extraire les titres des articles scientifiques. Une approche basée sur des règles est proposée par [Alamoudi et al., 2021] pour extraire les métadonnées des livres PDF. Dans le domaine financier, [EL-Haj et al., 2014], [EL-Haj et al., 2019b] et [EL-Haj et al., 2019a] ont proposé des approches sur les traitements des rapports annuels financier PDF en fonction de la récupération de texte et de la détection de tableau. Certains outils tels que pdftitle [8] et pdftotext [9] fournissent des informations sur la structure mais ils ne peuvent pas la détecter eux-même. L'outil pdfact développé par [Bast and Korzen, 2013] pour l'icecite, un système de gestion des documents de recherche (RPMS), peut extraire les rôles sémantiques tels que les titres, l'auteur, l'année, les blocs de texte et les pieds de page. Il fonctionne sur deux étapes : 1) l'extraction des caractères et l'identification des candidats des titres avec un score calculé basé sur les indices de ligne, la taille de police, et l'existence de format gras ou

8. https ://github.com/metebalci/pdftitle
9. https ://www.xpdfreader.com/

italique est calculé ; 2) appariement avec les candidats avec des jeux de références DBLP [10] et PubMed [11] .

En outre, [Ramakrishnan et al., 2012] proposent une approche en trois étapes qui extrait des blocs de texte à partir d'articles biologiques de recherche en texte intégral au format PDF et les classe en unités logiques en fonction de règles qui caractérisent des sections spécifiques : 1) la détection de blocs de texte contigus ; 2) la classification des blocs de texte en catégories rhétoriques ; 3) le regroupement des blocs de texte classifiés dans le bon ordre.

1. La **détection de blocs de texte contigus** commence par la détection des « blocs de mots » (délimitation des mots) avec la librairie Java JPedal afin de créer construire des « chunk-blocs » de texte tout en respectant les contraintes de formatage telles que le formatage à deux colonnes ou à une colonne. L'algorithme d'identification des blocs de texte fonctionne en regroupant des blocs de mots suffisamment proches (basé sur les statistiques spatiales des mots sur la page) et en se basant également sur des caractéristiques de police d'écriture.
2. La **classification des blocs de texte en catégories rhétoriques** basée sur « DROOLS », un système de gestion de règles métier et une implémentation du moteur de règles « ReteOO », basée sur l'algorithme Rete [Forgy, 1990] adapté au langage Java distribué dans le cadre de l'open-source JBoss Enterprise Platform [12]. Ils ont construit des fichiers de règles dans l'ensemble de données biologiques.
3. La dernière étape du **regroupement des blocs de texte classifiés** dans le bon ordre est d'extraire avec précision le texte de toute section donnée dans le bon ordre avec des titres de section et de sous-section correctement délimités.

A la fin, ils ont effectué une évaluation pour chacune de ces étapes. Pour la première étape, ils ont d'abord segmenté manuellement chaque page de leur jeu de données pour obtenir une segmentation idéale de chaque article. Ensuite, ils ont compté le nombre d'opérations d'édition (suppression et ajout de blocs de texte) nécessaires pour transformer les papiers segmentés manuellement en une segmentation prévue par l'outil. Puis, les mesures d'évaluation de précision, rappel et F-mesure ont été adoptées pour la deuxième étape. Ils ont calculé la reconnaissance de titre (Heading) et de bloc de texte (body) séparément. Enfin, pour la troisième étape, ils ont comparé leur approche avec l'outil pdftotext à l'aide d'une variante de l'algorithme Needleman-Wunsch[Needleman, 1970] et leur approche a eu une précision plus élevée que l'outil pdftotext.

2.5 Conclusion

Au terme de ce tour d'horizon de l'état de l'art, nous pouvons retenir plusieurs éléments essentiels concernant les différents types de documents PDF et les méthodes et outils existants pour l'extraction de texte et la détection de la structure. D'abord, il existe trois types de documents PDF qui impliquent chacun différentes méthodes de traitement. Donc, il est nécessaire de connaître le type de document avant de choisir

10. http ://dblp.uni-trier.de/
11. http ://www.ncbi.nlm.nih.gov/pubmed/
12. http ://labs.jboss.com/portal/jbossrules/
12. http ://labs.jboss.com/portal/jbossrules/

les outils pertinents. Ensuite, divers outils open source ou commerciaux nous permettent de récupérer les données que nous souhaitons.

Deuxième partie

Expérimentations

CHAPITRE 3

PRÉPARATION DE DATASETS

Sommaire

3.1 Introduction

Dans ce chapitre, nous allons présenter, dans un premier temps, les documents Offering Memorandum (OM) utilisés pour ce projet. Ensuite, nous allons montrer la construction d'un jeu de données composé de trois corpus. Nous avons dû construire un corpus pour chaque expérimentation décrite dans le chapitre 4 en prenant en compte les besoins de formats différents. Nous allons détailler les étapes de la construction du corpus.

3.2 Description de type de documents Offering Memorandum

Un Offering Memorandum (OM), également connu sous le nom de *private placement memorandum* (PPM), est un document juridique qui décrit les objectifs, les risques et les conditions d'un investissement impliqué dans un placement privé[Hayes, 2020]. Ce document comprend des informations clés sur la stratégie de croissance future de l'entreprise, les opportunités à venir sur le marché, les différentes stratégies utilisées pour réaliser les projections futures et des détails sur la concurrence du marché. La manière dont l'équipe de direction actuelle prévoit de traiter les faiblesses, l'évolutivité des opérations, etc., est détaillée dans le document. Un OM peut être écrit dans toutes les langues et les documents PDF utilisés pour ce projet sont en anglais.

Chaque OM sera personnalisé en fonction de l'investissement, mais chacun doit inclure certaines informations détaillées pour garantir que les investisseurs disposent de toutes les informations dont ils ont besoin. En général, un document OM

contient les sections suivantes [1] afin de présenter une image réaliste du secteur dans lequel l'entreprise opère et montrer clairement à l'investisseur quelles sont les perspectives et les objectifs de l'entreprise.

— Résumé de l'offre (Summary of the Offering)
— Résumé de l'activité (Business Summary)
— Exigences pour les acheteurs (Requirements for Purchasers) :
— Informations prospectives (Forward-Looking Information) :
— Facteurs de risque (Risk Factors)
— Utilisation des produits (Use of Proceeds)
— La gestion (Management)
— Compensation (Compensation)
— Conseil d'administration (Board of Directors)
— Tableau de capitalisation et dilution (Capitalization Table and Dilution)
— Information légale (Legal Information)

Les documents OM sont composés principalement de texte, mais ils contiennent également des tableaux qui permettent de présenter les informations plus clairement. En termes de structure, la mise en page a souvent une structure à deux colonnes, de sorte qu'il est plus pratique et plus clair de présenter le contenu avec des sous-sections. Les informations mentionnées dans le résumé seront présentées et expliquées en détail dans les chapitres correspondants suivants. Le nombre de pages d'un document OM peut varier de 50 à 500 pages, voire plus. Donc, pour un document avec un si grand nombre de pages, il serait très coûteux de trouver les informations nécessaires sur l'investissement manuellement. Un système d'extraction d'informations peut améliorer l'efficacité et trouver les informations nécessaires plus rapidement.

3.3 L'évaluation en trois étapes

Globalement, le système d'extraction de texte extrait tout d'abord le texte (les caractères et les espaces), en même temps que l'extraction également d'autres informations complémentaires telles que les coordonnées, les formats des caractères, les polices, les couleurs et la taille. A partir de ces données (Figure 3.1), il est capable de détecter la structure d'un document PDF en identifiant les titres et son bloc de texte correspondant. Les paires des titres et ses blocs de texte permettent aux utilisateurs de l'application de spécifier où extraire les informations qu'ils souhaitent.

En conséquence, nous avons divisé l'évaluation en trois étapes afin de mieux comprendre comment se déroule la reconnaissance des textes : 1) l'évaluation sur l'extraction de texte ; 2) l'évaluation de la détection des formats de caractères ; et 3) 3) l'évaluation de l'extraction de la structure.

— L'évaluation de l'extraction de texte nous permet de comprendre si les outils ont reconnu correctement les mots en suivant l'ordre des mots et paragraphes. Cinq outils d'extraction du texte ont été utilisés pour la première évaluation comparative : pdftotext, pdfact, FineReader PDF (Abbyy), pdfplumber, et pdfExtract (Fortia) ;
— Les formats sont des éléments importants pour l'identification des titres ou des blocs de texte. Ainsi, l'évaluation de la détection de format de caractères nous permet de distinguer dans un premier temps si les outils sont capables

1. https ://corporatefinanceinstitute.com/resources/knowledge/deals/offering-memorandum/

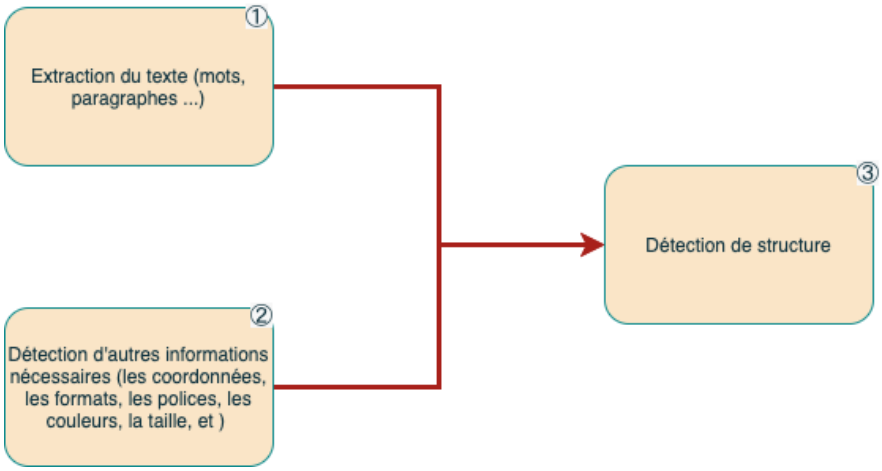


FIGURE 3.1 – Trois étapes de la reconnaissance des textes

de détecter les formats et ensuite la performance de cette détection. Les trois outils qui ont été utilisés pour cette deuxième évaluation comparative : pdfact, FineReader PDF, et DocParser (Fortia) ;

— La détection de structure de titres et de ses blocs de texte, comme présenté précédemment, dépend fortement des résultats de la reconnaissance de texte et de l'analyse heuristique sur les résultats de la détection de format. Deux outils ont été utilisés pour l'évaluation de l'extraction de la structure des documents : pdfact et docParser.

Pour chacune de ces évaluations, les outils utilisés pour les comparaisons des résultats extraient le même contenu. Après cette extraction, une comparaison est faite avec la sortie de l'outil. Pour cela, un ensemble de critères a été adopté et établi permettant une évaluation des outils d'extraction en comparant les résultats des outils avec le corpus de référence.

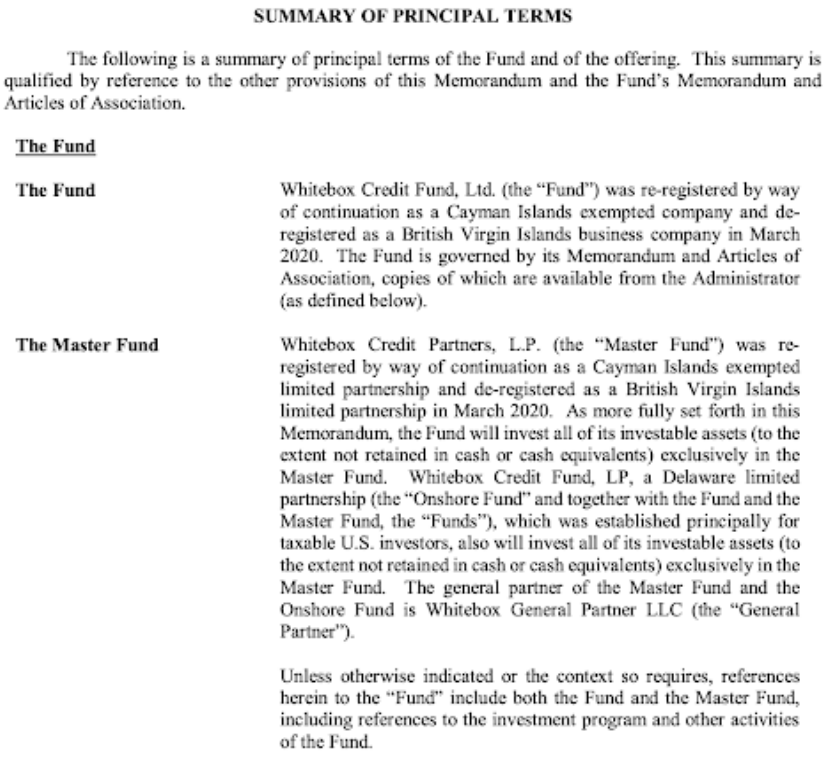
3.4 Construction de jeux de données

Notre jeux de données est composé de 32 documents PDF OM pour un ensemble total de 1,857 pages et de 955,926 mots. A la différence des informations utilisées pour l'évaluation dans chaque expérience, nous avons construit un corpus de référence pour chaque expérience bien que ce soit un travail long et coûteux, et qui nécessite beaucoup d'efforts manuels[Strecker et al., 2009]. Ce jeu de données n'inclut ni les données de tableaux ni les images car ces documents financiers se composent majoritairement de textes.

3.4.1 Construction de corpus pour la conversion en texte

Les outils utilisés pour la comparaison sont pdftotext, pdfact, FineReader PDF (Abbyy), pdfplumber, et pdfExtract (Fortia). Nous avons d'abord récupéré les résultats de sortie de la conversion en texte de 32 documents PDF qui sont de type natif avec ces outils sous forme de texte brut car pour une page du document OM, les fichiers de sortie des outils sont différents (Figure 3.2).

Nous pouvons observer dans la figure 3.2 que l'ordre de l'extraction du texte est différent selon les différents outils. Certains outils ont extrait le texte ligne par ligne



SUMMARY OF PRINCIPAL TERMS

The following is a summary of principal terms of the Fund and of the offering. This summary is qualified by reference to the other provisions of this Memorandum and the Fund's Memorandum and Articles of Association.

The Fund

The Fund	Whitebox Credit Fund, Ltd. (the "Fund") was re-registered by way of continuation as a Cayman Islands exempted company and de-registered as a British Virgin Islands business company in March 2020. The Fund is governed by its Memorandum and Articles of Association, copies of which are available from the Administrator (as defined below).
The Master Fund	Whitebox Credit Partners, L.P. (the "Master Fund") was re-registered by way of continuation as a Cayman Islands exempted limited partnership and de-registered as a British Virgin Islands limited partnership in March 2020. As more fully set forth in this Memorandum, the Fund will invest all of its investable assets (to the extent not retained in cash or cash equivalents) exclusively in the Master Fund. Whitebox Credit Fund, LP, a Delaware limited partnership (the "Onshore Fund" and together with the Fund and the Master Fund, the "Funds"), which was established principally for taxable U.S. investors, also will invest all of its investable assets (to the extent not retained in cash or cash equivalents) exclusively in the Master Fund. The general partner of the Master Fund and the Onshore Fund is Whitebox General Partner LLC (the "General Partner"). Unless otherwise indicated or the context so requires, references herein to the "Fund" include both the Fund and the Master Fund, including references to the investment program and other activities of the Fund.

FIGURE 3.2 – Exemple d'une page du document OM

```
256  <0x0b>SUMMARY OF PRINCIPAL TERMS
257  The following is a summary of principal terms of the Fund and of the offering. This summary is
258  qualified by reference to the other provisions of this Memorandum and the Fund's Memorandum and
259  Articles of Association.
260  The Fund
261  The Fund
262
263  Whitebox Credit Fund, Ltd. (the "Fund") was re-registered by way
264  of continuation as a Cayman Islands exempted company and deregistered as a British Virgin Islands business company in March
265  2020. The Fund is governed by its Memorandum and Articles of
266  Association, copies of which are available from the Administrator
267  (as defined below).
268
269  The Master Fund
270
271  Whitebox Credit Partners, L.P. (the "Master Fund") was reregistered by way of continuation as a Cayman Islands exempted
272  limited partnership and de-registered as a British Virgin Islands
273  limited partnership in March 2020. As more fully set forth in this
274  Memorandum, the Fund will invest all of its investable assets (to the
275  extent not retained in cash or cash equivalents) exclusively in the
276  Master Fund. Whitebox Credit Fund, LP, a Delaware limited
277  partnership (the "Onshore Fund" and together with the Fund and the
278  Master Fund, the "Funds"), which was established principally for
279  taxable U.S. investors, also will invest all of its investable assets (to
280  the extent not retained in cash or cash equivalents) exclusively in the
281  Master Fund. The general partner of the Master Fund and the
282  Onshore Fund is Whitebox General Partner LLC (the "General
283  Partner").
284  Unless otherwise indicated or the context so requires, references
285  herein to the "Fund" include both the Fund and the Master Fund,
286  including references to the investment program and other activities
287  of the Fund.
```

FIGURE 3.3 – Texte extraite par pdftotext ligne par ligne

```
118  SUMMARY OF PRINCIPAL TERMS
119
120  The following is a summary of principal terms of the Fund and of the offering. This summary is qualified by reference to the other
     provisions of this Memorandum and the Fund's Memorandum and Articles of Association.
121
122  The Fund
123
124  The Fund Whitebox Credit Fund, Ltd. (the "Fund") was re-registered by way of continuation as a Cayman Islands exempted company and
     de-registered as a British Virgin Islands business company in March 2020. The Fund is governed by its Memorandum and Articles of
     Association, copies of which are available from the Administrator (as defined below).
125
126  The Master Fund Whitebox Credit Partners, L.P. (the "Master Fund") was re-registered by way of continuation as a Cayman Islands
     exempted limited partnership and de-registered as a British Virgin Islands limited partnership in March 2020. As more fully set forth
     in this Memorandum, the Fund will invest all of its investable assets (to the extent not retained in cash or cash equivalents)
     exclusively in the Master Fund. Whitebox Credit Fund, LP, a Delaware limited partnership (the "Onshore Fund" and together with the
     Fund and the Master Fund, the "Funds"), which was established principally for taxable U.S. investors, also will invest all of its
     investable assets (to the extent not retained in cash or cash equivalents) exclusively in the Master Fund. The general partner of the
     Master Fund and the Onshore Fund is Whitebox General Partner LLC (the "General Partner").
127
128  Investment Objectives and Strategies
129
130  Unless otherwise indicated or the context so requires, references herein to the "Fund" include both the Fund and the Master Fund,
     including references to the investment program and other activities of the Fund.
```

FIGURE 3.4 – Texte extraite par pdfact paragraphe par paragraphe

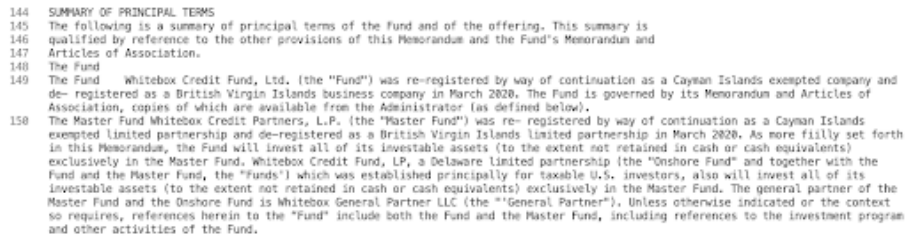


FIGURE 3.5 – Texte extraite par FineReader PDF paragraphe par paragraphe

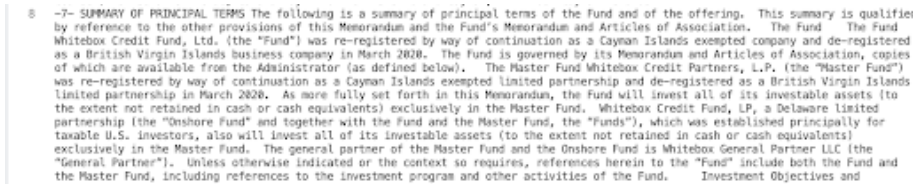


FIGURE 3.6 – Texte extraite par pdfplumber page par page

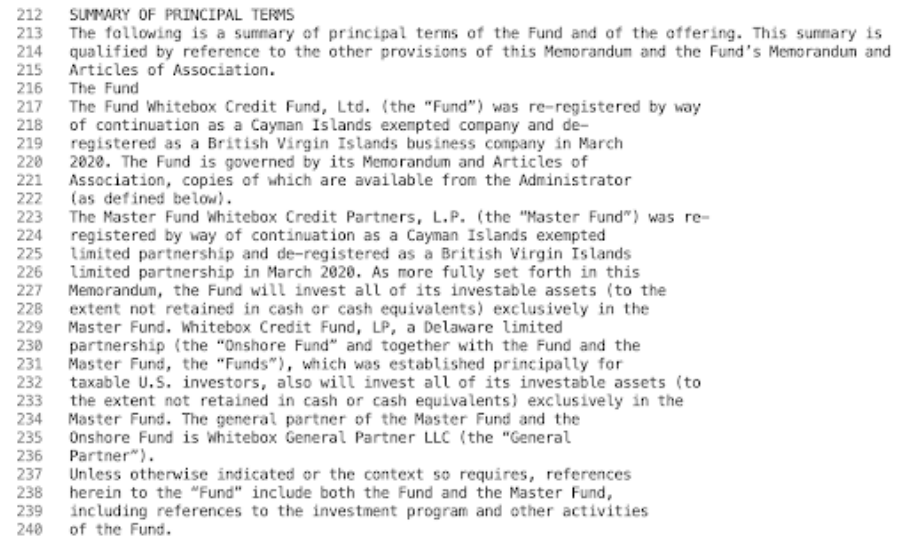


FIGURE 3.7 – Texte extraite par pdfExtract ligne par ligne

(Figures 3.3 et 3.7), alors que certains l'ont fait paragraphe par paragraphe (Figures 3.4 et 3.5). De plus, pdfplumber a extrait le texte page par page (Figure 3.6).

Le corpus de référence construit manuellement pour l'évaluation de la conversion en texte est également sous format de texte brut en suivant l'ordre de lecture et l'ordre des paragraphes.

L'ordre de lecture de la page suit d'abord le titre « SUMMARY OF PRINCIPAL TERMS » de la section et ensuite le paragraphe de l'introduction et continue enfin de la gauche vers la droite pour chaque sous-section selon le contexte. Le fichier de référence a suivi cet ordre (Figure 3.8). Nous avons traité les textes paragraphe par paragraphe et les avons mis sur une ligne afin que cela soit plus logique en termes de lecture. S'il existe une ligne blanche entre deux paragraphes, nous avons également mis une ligne vide.

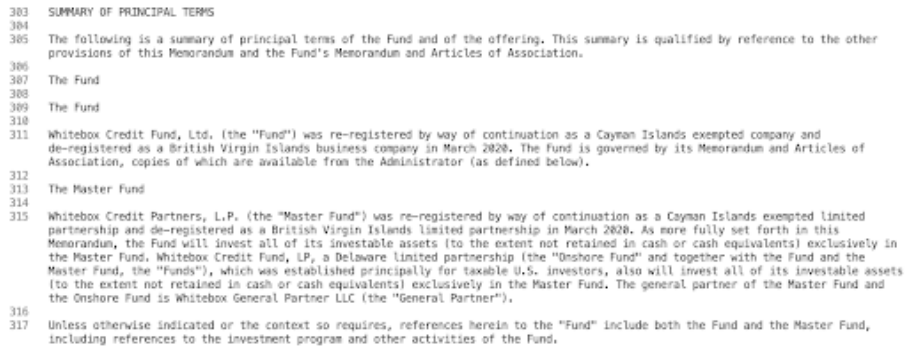


FIGURE 3.8 – Données de référence faites manuellement

3.4.2 Construction de corpus pour la détection du format de caractère

Les outils utilisés pour la comparaison de la détection du format de caractère sont FineReader PDF (Abbyy) et pdfExtract (Fortia). Les fichiers de sortie de FineReader PDF qui contiennent les informations sur le format de caractère sont des fichiers HTML alors que les fichiers de sortie de pdfExtract sont des fichiers JSON. Nous pouvons également noter comme différence majeure entre ces deux outils le fait que les informations extraites à l'aide de pdfExtract se font au niveau de la lettre alors que le FineReader PDF extrait des informations au niveau des mots ou de la phrase.

Nous pouvons observer dans les résultats HTML générés par FineReader PDF ci-dessous que les informations de format de caractère sont autour du texte avec l'étiquetage ***span*** dont les attributs ***classe*** et ***style*** définissent la mise en forme des textes.

```
<h1><a name="bookmark1"></a><span class="font2"
    style="font-weight:bold;"><a name="bookmark22"></a>SUMMARY OF
    PRINCIPAL TERMS</span></h1>
```

En revanche, le pdfextract extrait les informations lettre par lettre tels que les coordonnées, la police, la taille et la couleur.

```
        {
          "bounds": {
            "left": 331.82,
            "top": 393.46,
            "width": 4.78,
            "height": 7.46
          },
          "text": "S",
          "font": {
            "size": 10,
            "name": "TimesNewRomanPS-BoldMT",
            "color": "ff000000"
          }
        }
```

En conséquence, nous avons dû modifier les fichiers de sortie de ces deux outils au niveau du mot et ensuite les convertir au format JSON (Annexe A.2). Nous avons également récupéré les informations clés pour la comparaison :

— bold : est-ce que le token courant est en gras ?
— italique : est-ce que le token courant est en italique ?
— souligné : est-ce que le token courant est en souligné ?

Si la valeur correspondante est égale à 1, cela indique que selon l'outil, le token courant est sous ce format, alors que si la valeur correspondante est égale à 0, cela veut dire que soit le token courant n'est pas sous ce format, soit l'outil n'est pas capable de détecter ce format.

Nous avons décidé de prendre en compte 4,210 titres et sous titres de sections qui sont des titres uniques dans son document correspondant pour la comparaison. Le corpus de référence est sous un format JSON similaire (Annexe A.2).

3.4.3 Construction de corpus pour la détection de la structure

La structure d'un document PDF contient plusieurs rôles sémantiques : les titres du document, titres de section, les blocs de texte correspondant à un titre, les en-têtes et les pieds de page.

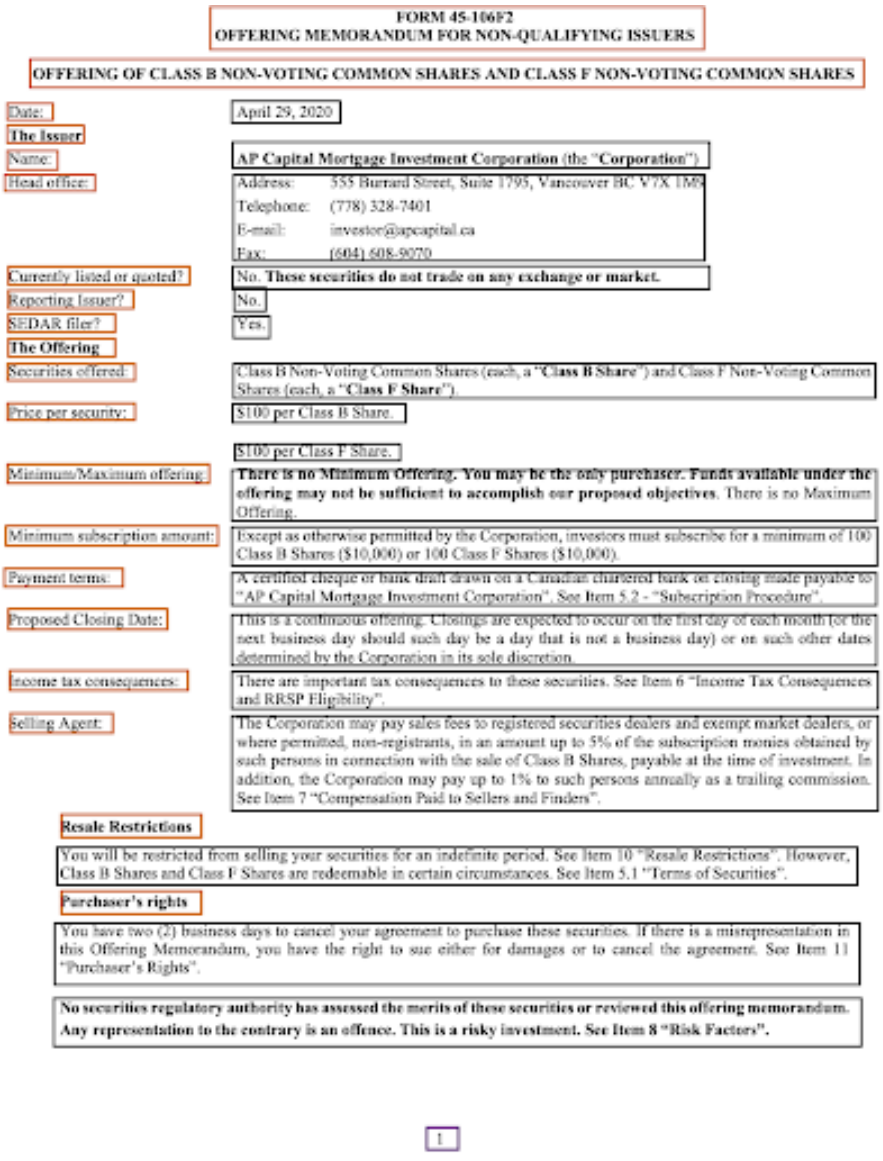
Comme présenté dans le chapitre 1, les données de titre et ses blocs de texte correspondants sont adoptés pour préciser les bons endroits où extraire des informations. C'est également une fonctionnalité de l'application DOC Reader qui permet aux utilisateurs de préciser les titres des sections afin de trouver des informations dont ils ont besoin. Par exemple, dans la figure 3.9, le titre « Date : » a un bloc de texte associé : « April 29, 2020 » alors que le titre « The Issuer » n'a pas de bloc de texte correspondant. Ainsi, la détection de titres et leurs blocs de texte correspondants joue un rôle important pour pouvoir ensuite se focaliser sur des endroits précis des PDF lors de l'extraction d'informations. Par exemple, si l'utilisateur souhaite extraire des informations sur le champ « minimumSubscription » qui est le montant de souscription minimale pour investir dans un fond, il peut ajouter « minimum subscription » comme titre dans la configuration de l'application. L'application va d'abord aller chercher les titres contenant le mot-clé « minimum subscription » pour ensuite détecter le titre « Minimum subscription amount : » sur la page de l'exemple (Figure 3.9) et finalement extraire l'information recherchée dans le bloc de texte correspondant pour le champ « minimumSubscription » grâce à des modèles d'apprentissage automatique. En conséquence, la performance de la détection de la structure impacte les informations extraites par les systèmes d'extraction d'informations.

Les outils utilisés pour l'évaluation sont l'outil pdfact de l'icecite et le docParser de Fortia. Le docParser, basé sur les informations extraites par le pdfExtract, est un outil propriétaire de Fortia et il a pour objectif de récupérer les textes des titres et des blocs de texte.

Avant de convertir les données dans un format JSON exploitable, nous avons remarqué que leurs étiquettes sur la détection de la structure ne sont pas identiques : (Tableau 3.1) :

DocParser	**Pdfact**
titre	title (titre du document)
	heading (titre de la section)
bloc de texte	body (bloc de texte)
	footer
(bloc de texte)	other

TABLE 3.1 – Étiquettes différentes de structure de ces deux outils



FORM 45-106F2
OFFERING MEMORANDUM FOR NON-QUALIFYING ISSUERS

OFFERING OF CLASS B NON-VOTING COMMON SHARES AND CLASS F NON-VOTING COMMON SHARES

Date:	April 29, 2020
The Issuer	
Name:	**AP Capital Mortgage Investment Corporation** (the "**Corporation**").
Head office:	Address: 555 Burrard Street, Suite 1795, Vancouver BC V7X 1M9 Telephone: (778) 328-7401 E-mail: investor@apcapital.ca Fax: (604) 608-9070
Currently listed or quoted?	No. **These securities do not trade on any exchange or market.**
Reporting Issuer?	No.
SEDAR filer?	Yes.
The Offering	
Securities offered:	Class B Non-Voting Common Shares (each, a "**Class B Share**") and Class F Non-Voting Common Shares (each, a "**Class F Share**").
Price per security:	$100 per Class B Share. $100 per Class F Share.
Minimum/Maximum offering:	**There is no Minimum Offering. You may be the only purchaser. Funds available under the offering may not be sufficient to accomplish our proposed objectives.** There is no Maximum Offering.
Minimum subscription amount:	Except as otherwise permitted by the Corporation, investors must subscribe for a minimum of 100 Class B Shares ($10,000) or 100 Class F Shares ($10,000).
Payment terms:	A certified cheque or bank draft drawn on a Canadian chartered bank or closing made payable to "AP Capital Mortgage Investment Corporation". See Item 5.2 - "Subscription Procedure".
Proposed Closing Date:	This is a continuous offering. Closings are expected to occur on the first day of each month (or the next business day should such day be a day that is not a business day) or on such other dates determined by the Corporation in its sole discretion.
Income tax consequences:	There are important tax consequences to these securities. See Item 6 "Income Tax Consequences and RRSP Eligibility".
Selling Agent:	The Corporation may pay sales fees to registered securities dealers and exempt market dealers, or where permitted, non-registrants, in an amount up to 5% of the subscription monies obtained by such persons in connection with the sale of Class B Shares, payable at the time of investment. In addition, the Corporation may pay up to 1% to such persons annually as a trailing commission. See Item 7 "Compensation Paid to Sellers and Finders".

Resale Restrictions

You will be restricted from selling your securities for an indefinite period. See Item 10 "Resale Restrictions". However, Class B Shares and Class F Shares are redeemable in certain circumstances. See Item 5.1 "Terms of Securities".

Purchaser's rights

You have two (2) business days to cancel your agreement to purchase these securities. If there is a misrepresentation in this Offering Memorandum, you have the right to sue either for damages or to cancel the agreement. See Item 11 "Purchaser's Rights".

No securities regulatory authority has assessed the merits of these securities or reviewed this offering memorandum. Any representation to the contrary is an offence. This is a risky investment. See Item 8 "Risk Factors".

1

FIGURE 3.9 – Exemple de la structure d'une page de document OM : les rectangles oranges sont des titres, les rectangles noires sont des bloc des texte et le rectangle violet est un pied de page

DocParser a deux étiquettes (titre et bloc de texte) en filtrant les en-têtes et les pieds de page car ces informations ne seront pas utilisées pour entraîner les modèles. L'outil pdfact a cinq étiquettes : title, heading, body, footer et other. Nous avons donc traité le *title* et le *heading* de pdfact comme des titres et *body* et *other* comme des blocs de texte afin qu'ils soient homogènes pour la comparaison.

Le nombre de paires de titres et de blocs de texte correspondant pour l'évaluation est de 4,716. Tous les fichiers de sortie et les fichiers de référence sont en format JSON et contiennent le titre, le bloc de texte et le numéro de la page auquel il appartient et l'id :


```
{
    "title": {
      "text": [
        "The Fund "
      ]
    },
    "page": 7,
    "body": [
      "Whitebox Credit Fund, Ltd. (the "Fund") was
         re-registered by way of continuation as a Cayman
         Islands exempted company and de- registered as a
         British Virgin Islands business company in March 202
         0. The Fund is governed by its Memorandum and
         Articles of Association, copies of which are
         available from the Administrator (as defined
         below)."
    ],
    "id": 26
  }
```


3.5 Conclusion

Nous avons d'abord présenté les documents OM qui ont été pris en compte pour la construction de jeux de données. Les documents OM sont très structurés en termes de contenu mais très personnalisés au niveau de la structure. Cela implique des difficultés au cours de l'extraction de texte. Ensuite, nous avons divisé l'évaluation en trois étapes et présenté les outils utilisés pour chaque étape . Enfin, nous avons montré la préparation et la construction des corpus pour les trois niveaux de l'évaluation. En effet, nous avons construit un corpus pour chaque niveau de l'évaluation en raison des différentes données utilisées pour la comparaison.

CHAPITRE 4

EVALUATION DES OUTILS POUR CONVERTIR EN TEXTES ET RÉCUPÉRER D'AUTRES INFORMATIONS À PARTIR DE DOCUMENTS PDF

Sommaire

4.1 Introduction

Dans ce chapitre, nous allons présenter les évaluations de la conversion de PDF en texte avec cinq outils, pdftotext, pdfact, pdfExtract de Fortia, FineReader PDF de l'Abbyy, et pdfplumber mentionnés dans le chapitre 3. Ensuite, nous allons évaluer la détection des formats de caractères avec deux outils pdfExtract et FineReader PDF et à la fin nous allons montrer l'évaluation et les résultats de la détection de structure de PDF avec deux outils, docParser de Fortia et pdfact de l'icecite.

4.2 Résultats de la conversion de PDF en texte

4.2.1 Méthode d'évaluation

L'évaluation de la conversion de PDF en texte est composée de trois parties : au niveau du mot, de la ligne et du paragraphe. Au niveau des mots, nous examinons si les textes produits contiennent des mots hors vocabulaire en raison de caractères mal reconnus. Au niveau de lignes, nous analysons principalement si les outils détectent

la frontière des paragraphes. Il est possible qu'un outil ait bien détecté la frontière des mots avec peu de caractères mal reconnus. En revanche, il peut détecter un paragraphe de trois lignes en trois lignes séparées au lieu d'une seule ligne. Au niveau du paragraphe, nous souhaitons analyser si l'ordre de lecture a été correctement représenté dans la sortie.

Ainsi, nous avons adopté la méthode de [Bast and Korzen, 2017] qui propose une approche d'évaluation de résultats de conversion de texte. Le travail propose les métriques ci-dessous comme critère d'évaluation :

- Au niveau de la reconnaissance des mots et son frontière :
  - W + : le nombre de mots erronés dans le fichier de sortie.
  - W - : le nombre de mots manquants dans le fichier de sortie.
  - W $\sim$: le nombre de mots mal orthographiés dans le fichier de sortie.
- Au niveau de la détection des frontières de paragraphe :
  - NL + : le nombre de sauts de ligne erronés dans le fichier de sortie.
  - NL - : le nombre de sauts de ligne manquants dans le fichier de sortie.
- Au niveau de la distinction entre les paragraphes et de l'ordre de lecture :
  - P + : le nombre de paragraphes erronés dans le fichier de sortie.
  - P - : le nombre de paragraphes manquants dans le fichier de sortie..
  - P $\uparrow\downarrow$: paragraphes réarrangés dans le fichier de sortie.

Ils ont développé un algorithme heuristique *doc-diff* qui consiste à comparer les mots d'un fichier de sortie O et d'un fichier de référence G.

Ficher de sortie O	Fichier de référence G
as segregated portfolio company. <ligne blanche> FA SPC is incorporated in the Cayman <ligne blanche> Islands as an exempted company with <ligne blanche> limited liability registered	FA SPC is incorporated in the Cayman Islands as an exempted company with limited liability registered as a segregated portfolio company.

TABLE 4.1 – Exemple de fichiers de sortie et de référence

Pour l'exemple donné (Tableau 4.1), après avoir effectué un pré-traitement de fichier de référence G où les mots sont convertis en minuscules et toutes les ponctuations ont été enlevées, nous obtenons le résultat Fichier de sortie O à gauche. Nous supposons que w_O et w_G sont des listes de mots par paragraphe dans les fichiers O et G respectivement. Par conséquent, w_O est égale à [[as, segregated, portfolio, company], [fa, spc, is, incorporated, in, the, cayman], [islands, as, an, exempted, company, with],[limited, liability, registered]] et w_G est égale à [[fa, spc, is, incorporated, in, the, cayman, islands, as, an, exempted, company, with, limited, liability, registered, as, a, segregated, portfolio, company]], où chaque liste à l'index i contient les mots du paragraphe i.

L'approche *doc-diff* est utilisée afin de trouver les différences entre deux chaînes de mots et de les catégoriser dans les types suivants :

- Chaînes communes(=[mot_1, ..., mot_i]) : une séquence de i mots consécutifs communs à w_O et w_G .
- Chaînes différentes($\sim$ [mot_1, ..., mot_j], [mot_1, ..., mot_k]) : une séquence de j mots erronés, qui apparaissent dans w_O mais pas dans w_G ; et de k mots manquants, qui apparaissent dans w_G mais pas dans w_O .

— Chaînes réarrangées (↑↓ [mot$_1$,...,mot$_m$], [mot$_1$,...,mot$_n$]) : une séquence de m mots dans w$_O$ et n mots dans w$_G$, qui sont (presque) égaux (m n), mais leurs positions dans w$_O$ et w$_G$ ne sont pas corrélées.

Les chaînes sont calculées en deux étapes.

À la première étape, les chaînes communes et différentes sont calculées par un algorithme appelé *word-diff* qui fonctionne de manière similaire à la commande *diff* sous Unix, mais basée sur des mots au lieu de lignes. Par conséquent, pour l'exemple (Tableau 4.1), *word-diff* calcule les chaînes c1 :(∼[as segregated, portfolio, company], []) ; c2 : (= [fa, spc, is, incorporated, in, the, cayman]) ; c3 : (= [islands, as, an, exempted, company, with]) ; c4 : (= [limited, liability, registered]) ; et c5 : (∼ [], [as, a, segregated, portfolio, company]).

Dans la deuxième étape, les chaînes réarrangées sont calculées par un algorithme appelé *rearr-diff* qui est un algorithme d'alignement et fonctionne de manière similaire à l'algorithme de Smith-Waterman[Smith et al., 1981], mais basé sur des mots au lieu des caractères. Globalement, le *rearr-diff* examine les différentes chaînes, identifie les endroits où il y a des mots similaires entre les mots erronés et manquants, les met dans des chaînes réarrangées et associe les chaînes réarrangées avec les différentes chaînes associées. Par conséquent, pour l'exemple (Tableau 4.1), le *rearr-diff* identifie les endroits où les mots sont similaires entre les mots erronés de la chaîne c1 et les mots manquants de la chaîne c5 et crée la chaîne réarrangée c6 : (↑↓ [as segregated, portfolio, company], [as, a, segregated, portfolio, company]). Ces critères sont en effet facilement interprétables et indépendants.

Pour chaque chaîne de p$_i$ (p étant l'acronyme de paragraphe), *doc-diff* analyse les numéros de paragraphe de p$_i$ et p$_{i-1}$ afin d'identifier les sauts de paragraphe dans w$_O$ et w$_G$. S'il y a un saut de paragraphe dans w$_O$ mais pas dans w$_G$, un NL+ est ajouté au critère de chaîne de p$_i$. De même, s'il y a un saut de paragraphe dans w$_G$ mais pas dans w$_O$, un NL- est ajouté. Pour l'exemple ci-dessus(Tableau 4.1), un NL+ est ajouté au critère de chaîne de p$_2$ et p$_3$, car il y a un saut de paragraphe entre p$_2$ et p$_3$ dans w$_O$,mais pas dans w$_G$. A la fin, l'affectation finale serait : P ↑↓ = 1, NL+= 3 ; W-= 1.

4.2.2 Résultats et analyses de la conversion de PDF en texte

Dans (Tableau 4.2) nous présentons les résultats de l'évaluation des outils d'extraction de texte dans les fichiers PDF pour chaque critère d'évaluation calculé par l'algorithme *doc-diff*.

Outil	W+	W-	W∼	NL+	NL-	P+	P-	P ↑↓
pdftotext	8,2 (0,0%)	13.2 (0,0%)	21,4 (0,1%)	100,7 (14.7%)	439,0 (63,9%)	34,8 (0,8%)	19,6 (0,2%)	3,8 (0,5%)
pdfact	**1,1 (0,0%)**	3,3 (0,0%)	**1,7 (0,0%)**	12,3 (1,8%)	**179,9 (26,2%)**	46,5 (0,9%)	34,7 (0,3%)	1,1 (0,1%)
pdfExtract	7,3 (0,0%)	1,5 (0,0%)	15,7 (0,1%)	4,0 (0,6%)	669,6 (97.5%)	**4,9 (0,7%)**	**11,0 (0,1%)**	3,0 (0,5%)
abbyy	10,8 (0,0%)	5,1 (0,0%)	208,4 (0,7%)	0,8 (0,1%)	684,3 (99.7%)	13,9 (0,9%)	16,8 (0,1%)	0,2 (0,0%)
pdfplumber	6,7 (0,0%)	**0,3 (0,0%)**	22,6 (0,1%)	**0,2 (0,0%)**	677,3 (98,6%)	36,0 (0,8%)	56,2 (0,3%)	**0,1 (0,0%)**

TABLE 4.2 – Résultats d'évaluation de 5 outils d'extraction PDF en texte

Nous observons que les taux d'erreur sur les mots (W+, W- et W) sont très faibles pour chacun des outils testés. Certains outils ont des problèmes de fautes d'orthographe en ajoutant un espace entre deux lettres d'un mot ou en oubliant un espace entre deux mots. Par exemple, l'outil pdfExtact a extrait la chaîne « of its » par « ofits ». De plus, Abbyy a comme autre problème d'identifier des lettres à la place d'autres

lettres ou à la place de nombres. Par exemple, il a extrait le mot « further » par « fiirther », et extrait le mot « IRS » par « 1RS ». C'est la raison pour laquelle il a une valeur légèrement plus élevée que les autres outils.

En ce qui concerne l'identification des frontières de paragraphe (NL+ et NL-), l'outil pdfact est le plus performant avec le taux d'erreur le moins élevé car il a extrait les textes paragraphe par paragraphe alors que les autres ont extrait les textes ligne par ligne. D'ailleurs, l'outil pdftotext a souvent délimité incorrectement les paragraphes commençant par un chiffre en deux paragraphes.

Enfin, tous les outils obtiennent des résultats similaires en matière de détection des paragraphes et de conservation de l'ordre de lecture.

4.3 Résultats de la détection des formats de caractères

4.3.1 Méthode d'évaluation

Certains outils sont capables de détecter la mise en forme des caractères (gras, italique, souligné), les majuscules, les polices et/ou les couleurs de caractères. Ce sont des caractéristiques essentielles pour l'identification de la structure d'un document, notamment, l'identification des frontières des titres. Les formats des caractères qui sont en gras, en italique et/ou soulignés sont souvent utilisés sur des titres pour qu'ils ressortent différemment du corps de texte.

Le tableau 4.3 présente les formats détectables par ces 5 outils, l'outil pdftotext ne fournit pas d'informations sur les formats, les couleurs ou les polices. En revanche, si nous lançons cet outil avec l'option "*–layout*", les résultats de sortie de pdftotext conservent les dispositions physiques telles que les indentations. En ce qui concerne les quatres autres outils, ils sont capables de détecter les informations de format (gras, italique et souligné) qui sont des critères pour cette évaluation.

Outil	Format (gras, italique, souligné)	Couleur	Police
pdftotext	NON	NON	NON
pdfact*	OUI	OUI	OUI
pdfExtract	OUI	OUI	OUI
abbyy	OUI	OUI	OUI
pdfplumber	OUI	OUI	OUI

TABLE 4.3 – Formats détectables selon les 5 outils, les fichiers de sortie par défaut de l'outil pdfact ne contient d'informations sur les formats. Néanmoins, nous avons réussi à les récupérer en changeant la format sortie en pdf.js.

L'évaluation a été effectuée selon ces critères sur les formats car les informations des codes des couleurs et des polices sont impossibles d'identifier à l'oeil :

— Est-ce que l'outil est capable de détecter les chaînes en gras ?
— Est-ce que l'outil est capable de détecter les chaînes en italique ?
— Est-ce que l'outil est capable de détecter les chaînes soulignées ?

Il existe 8 possibilités en termes de mise en forme : sans format, gras uniquement, italique uniquement, souligné uniquement, la combinaison de gras et souligné, la combinaison de gras et italique, la combinaison d'italique et souligné, et la combinaison des trois (gras, italique et souligné).

Dans le corpus de documents OM, nous observons la distribution suivante des mises en forme de titre : 2,491 titres uniquement en gras, 443 titres uniquement en

italiques et 25 titres uniquement en soulignées, 178 titres en gras et en italiques, 738 en gras et soulignés, 335 titres qui n'ont aucun format.

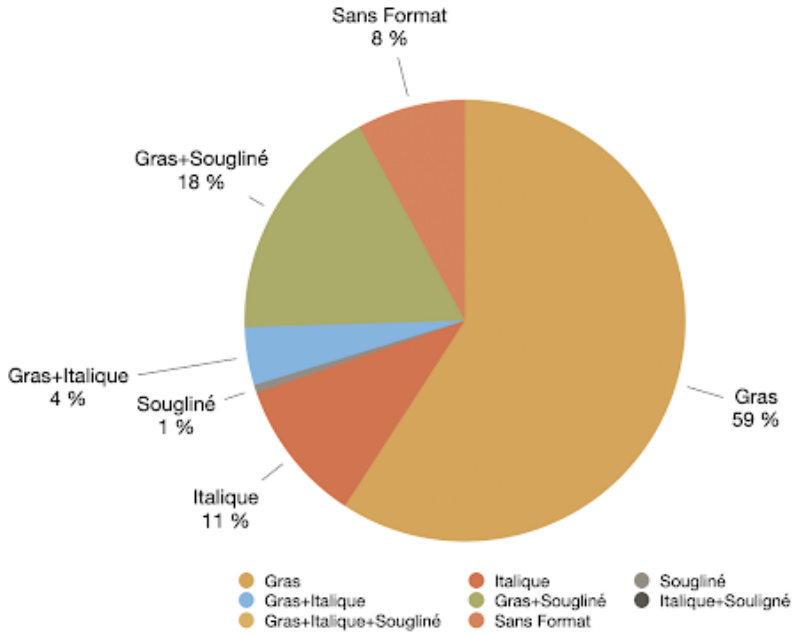


FIGURE 4.1 – Distribution des types de mise en forme des titres du corpus OM

La figure 4.1 présente la distribution des types de mises en forme des titres. Nous relevons que le gras est utilisé majoritaire seul, suivi du gras avec du souligné. Néanmoins, il n'existe pas de titres dans le corpus qui sont en italique et soulignés ou qui sont en gras et en italique et soulignés. Ainsi, ces deux possibilités ne seront pas analysées dans cette expérience. De plus, il est possible qu'un titre ne comporte aucun format (8%) mais ce n'est pas souvent le cas.

Après avoir mise en place tous les aspects pour l'évaluation, nous avons effectué l'évaluation en suivant la procédure décrite dans la figure 4.2 :

A partir du corpus de référence, nous avons d'abord récupéré les titres qui ont été correctement détectés par les outils. Cela nous a permis d'éviter les titres détectés ayant des problèmes de frontière de mots et de reconnaissance de caractères. Ensuite, nous avons comparé chaque titre avec les critères sélectionnés. Enfin, nous avons utilisé la précision (Formule 4.1), le rappel (Formule 4.2) et la F-mesure (Formule 4.3) comme mesures d'évaluation.

Nous distinguons :

— Les vrais positifs : ce qui est détecté comme ayant tel format par l'outil et qui est vrai ;
— Les faux positifs : ce qui est détecté comme ayant tel format par l'outil mais qui est faux ;
— Les faux négatifs : ce qui n'est pas détecté comme ayant tel format par l'outil mais qui est vrai

$$Précision = \frac{vrais\ positifs}{vrais\ positifs + faux\ positifs} \tag{4.1}$$

$$Rappel = \frac{vrais\ positifs}{vrais\ positifs + faux\ négatifs} \tag{4.2}$$

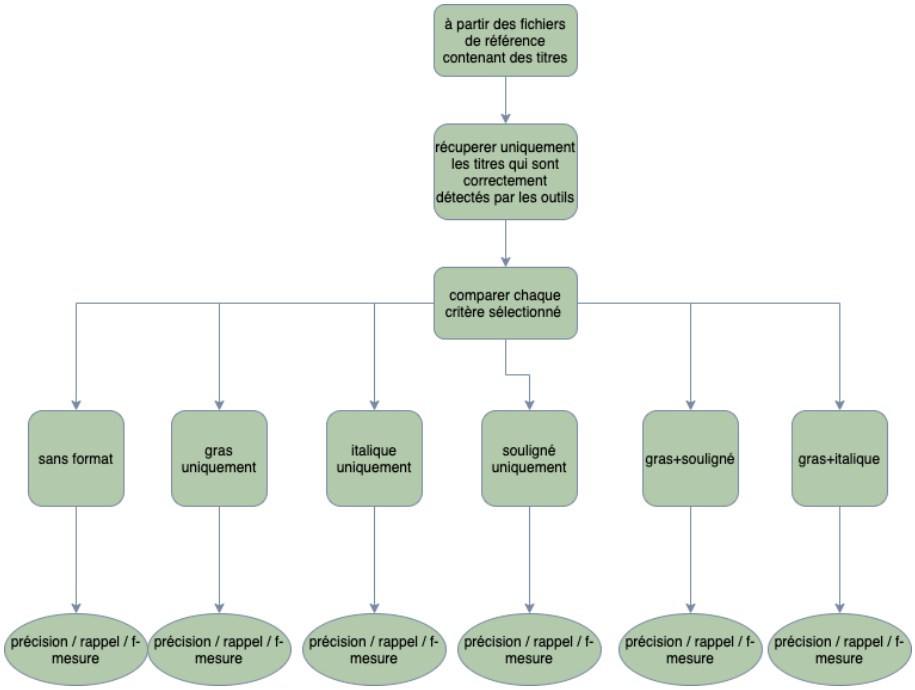


FIGURE 4.2 – Diagramme sur l'évaluation de la détection de format de caractère

$$F - mesure = \frac{(1+\beta^2) \times précision \times rappel}{\beta^2 \times précision + rappel} \tag{4.3}$$

4.3.2 Résultats et analyses de la détection des formats de caractères

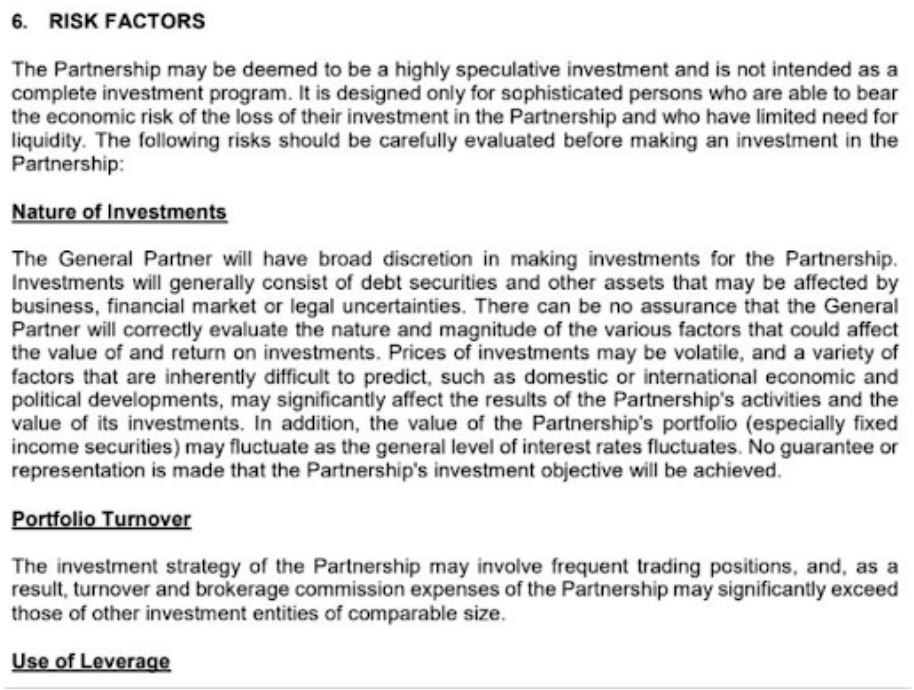
Parmi ces cinq outils sélectionnés, quatre outils (pdfact, pdfplumber, FineReader PDF, pdfExtract) sont capables d'extraire des formats de caractères. Ici, nous avons pris trois outils sauf l'outil pdfplumber car il a extrait des informations page par page au lieu de ligne par ligne et cela risque de ne pas localiser correctement des chaînes de titres. De plus, l'outil pdftotext ne fournit pas des informations concernant les formats de caractères.

Outil	sans format			gras uni			italique uni			souligné uni			gras+souligné			gras+italique		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
pdfExtract	0,78	1	0,88	0,68	0,98	0,81	0,99	1	**0,99**	0	0	0	0	0	0	0,98	0,88	0,92
FineReader PDF	0,96	0,87	**0,92**	1	0,99	**0,99**	0,79	0,97	0,87	0,82	1	0,90	0,99	0,99	**0,99**	0,97	1	**0,99**
pdfact	0,68	1	0,8	0,69	0,96	0,8	0,97	0,96	0,97	0	0	0	0	0	0	0,95	0,78	0,86

TABLE 4.4 – Résultats d'évaluation de 3 outils d'extraction des formats de caractère (uni=uniquement ; P=Précision ; R=Rappel ; F=F-mesure)

Ces trois outils sont assez performants pour les propriétés qu'ils sont capables de traiter. Nous pouvons remarquer que l'outil FineReader PDF n'est performant que pour 5 sur 6 possibilités analysées avec une F mesure assez élevée. Néanmoins, les performances des outils pdfEtract et pdfact sont très similaires. D'une part, avec les scores de 0 pour les formats soulignés uniquement et la combinaison de gras et

soulignés, ils ne sont pas capables de détecter les formats des chaînes sous ces formats comme dans les titres en gras soulignés de la figure 4.3.



6. RISK FACTORS

The Partnership may be deemed to be a highly speculative investment and is not intended as a complete investment program. It is designed only for sophisticated persons who are able to bear the economic risk of the loss of their investment in the Partnership and who have limited need for liquidity. The following risks should be carefully evaluated before making an investment in the Partnership:

Nature of Investments

The General Partner will have broad discretion in making investments for the Partnership. Investments will generally consist of debt securities and other assets that may be affected by business, financial market or legal uncertainties. There can be no assurance that the General Partner will correctly evaluate the nature and magnitude of the various factors that could affect the value of and return on investments. Prices of investments may be volatile, and a variety of factors that are inherently difficult to predict, such as domestic or international economic and political developments, may significantly affect the results of the Partnership's activities and the value of its investments. In addition, the value of the Partnership's portfolio (especially fixed income securities) may fluctuate as the general level of interest rates fluctuates. No guarantee or representation is made that the Partnership's investment objective will be achieved.

Portfolio Turnover

The investment strategy of the Partnership may involve frequent trading positions, and, as a result, turnover and brokerage commission expenses of the Partnership may significantly exceed those of other investment entities of comparable size.

Use of Leverage

FIGURE 4.3 – Titres en gras et en souligné

Nous pouvons observer que le format de caractère du titre « Nature of Investments » dans le texte est à la fois en gras et en souligné. Les outils pdfExtract et pdfact n'ont pas pu détecter les formats correctement alors que le FineReader PDF était capable de correctement détecter (Annexe A.3). En analysant les résultats en détail, nous avons pu observer que tous les 178 titres gras et soulignés n'ont pas été détectés correctement.

D'autre part, les performances sur la détection des autres formats sont également similaires avec une F-mesure faible pour la détection de format gras uniquement et la détection des titres sans format. La raison pour laquelle la précision du format « sans format » est faible est que les formats de soulignés non correctement détectés sont considérés comme « sans format ». Cela augmente le nombre de faux positifs pour ce critère. En revanche, l'outil pdfExtract est plus performant sur la détection de mots en italique uniquement.

4.4 Résultats de la détection de structure de PDF

4.4.1 Méthode d'évaluation

Cette évaluation a pour objectif de comparer si deux outils utilisés peuvent détecter la structure d'un document. La structure d'un document contient deux parties : titre et paragraphes. Comme indiqué dans le chapitre 1, les informations concernant la structure des titres seront utilisées pour localiser les sections adéquates où extraire les informations dans le système d'extraction d'informations du projet DOC Reader.

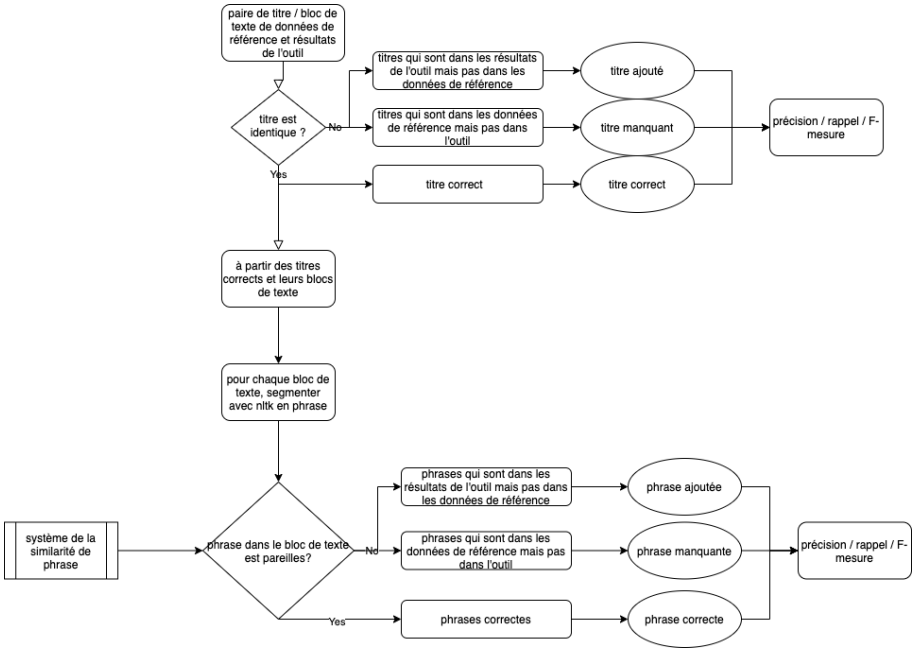


FIGURE 4.4 – Diagramme sur l'évaluation de la détection de structure

Le diagramme (Figure 4.4) montre que l'évaluation est réalisée en deux étapes : tout d'abord, nous avons comparé si un titre est correctement détecté à partir des paires de titres et de ses blocs de texte correspondants. Nous avons également utilisé la précision, le rappel et la F-mesure comme des mesures d'évaluation :

— Les vrais positifs sont les titres correctement détectés par l'outil (titre correct) ;
— Les faux positifs sont les titres détectés par l'outil mais qui ne sont pas des titres (titre ajouté) ;
— Les faux négatifs sont les titres qui n'ont pas été détectés par l'outil (titre manquant).

La deuxième étape d'évaluation porte sur les blocs de texte dont le titre est correctement détecté par l'outil. Nous avons segmenté en phrases chaque bloc de texte associé à un titre à l'aide de la librairie NLTK[1]. Ensuite, nous avons mesuré la similarité entre les phrases de la base données de référence et de l'outil afin de vérifier la similarité des phrases. Pour ce faire, nous avons fait appel au système de similarité de phrase de Fortia. Ce système est basé sur un modèle entraîné sur 1208 prospectus financiers en anglais, type de document financier similaire aux OM. Ce modèle de SIF embeddings (Smooth Inverse Frequency) calcule les embeddings de phrases comme une moyenne pondérée de vecteurs de mots [Arora et al., 2017] et la similarité cosinus donne la similarité entre deux vecteurs. Nous avons considéré que deux phrases étaient similaires si elles obtenaient un score de similarité supérieur à 0,8.

De même, nous avons mesuré les résultats de similarité de la même manière pour les paragraphes correspondant aux titres.

4.4.2 Résultats et analyses de la détection de structure de PDF

Nous pouvons observer que selon le tableau 4.5, l'outil docParser de Fortia a atteint une F-mesure de 0,84 sur la détection de titre, ce qui est plus performant que

1. https ://www.nltk.org/

pdfact qui a eu une F-mesure de 0,69. En terme de nombre de titres correctement détectés, l'outil docParser a réussi à détecter correctement 4,149 titres sur 4,716 titres alors que l'outil pdfact a détecté correctement 3,032 titres.

Outil	Précision	Rappel	F-mesure
docParser	0,81	0,88	**0,84**
pdfact	0,74	0,65	0,69

TABLE 4.5 – Résultat de l'évaluation de la détection de titre

La performance de la détection de structure est influencée non seulement par les méthodes de la détection ou l'algorithme du modèle, mais aussi par les bons résultats des étapes précédentes, notamment l'étape de la récupération de texte.

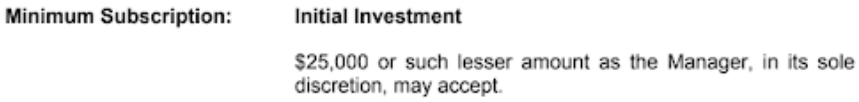


FIGURE 4.5 – Deux titres du document OM

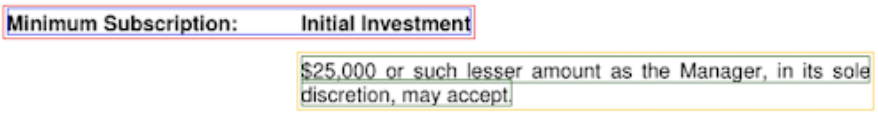


FIGURE 4.6 – La détection de titre par le docParser : ce qui dans le rectangle rouge est le titre détecté

Comme le montre l'exemple (Figure 4.5 et 4.6), nous remarquons que le résultat de la détection des deux titres (« Minimum Subscription : » et « Initial Investment ») est impacté par celui de l'outil pdfExtract qui a eu des soucis de délimitation sur la conversion en texte.

D'autre part, nous constatons que l'approche heuristique pour ce type de traitement montre certaines limites. Il est difficile de trouver les règles adaptable à toutes les situations. Dans l'exemple (Figure 4.7 et 4.8), nous pouvons voir que l'outil docParser a détecté la dernière ligne en gras du paragraphe comme un titre car dans les informations heuristiques, les lignes en gras sont considérées comme des titres.

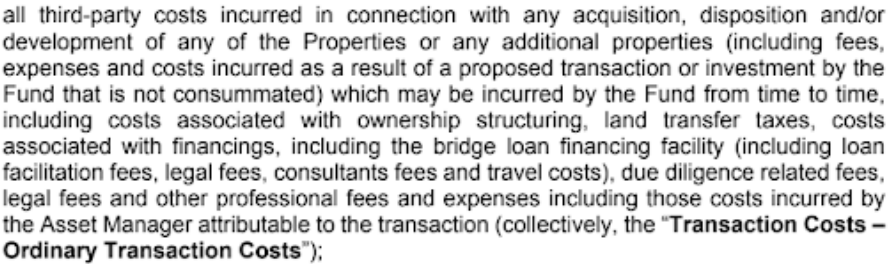

all third-party costs incurred in connection with any acquisition, disposition and/or development of any of the Properties or any additional properties (including fees, expenses and costs incurred as a result of a proposed transaction or investment by the Fund that is not consummated) which may be incurred by the Fund from time to time, including costs associated with ownership structuring, land transfer taxes, costs associated with financings, including the bridge loan financing facility (including loan facilitation fees, legal fees, consultants fees and travel costs), due diligence related fees, legal fees and other professional fees and expenses including those costs incurred by the Asset Manager attributable to the transaction (collectively, the "**Transaction Costs – Ordinary Transaction Costs**");

FIGURE 4.7 – Un paragraphe du document OM qui ne contient aucun titre

Pour l'outil pdfact, la limite principale est qu'il n'est pas capable de détecter des titres avec des structures en double colonne. La libraire JAVA PDFBox intégrée a eu des soucis de délimitation des frontières de paragraphes. Comme le montre l'exemple

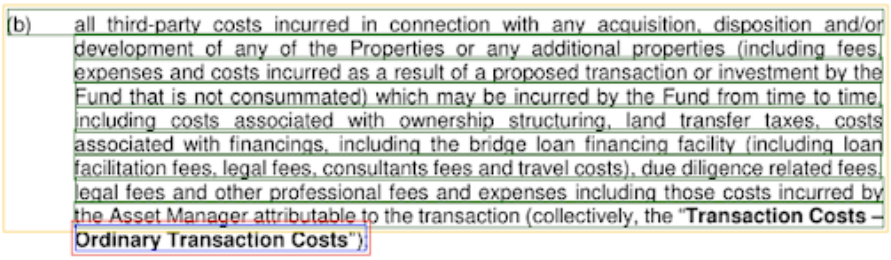


FIGURE 4.8 – La détection de titre par le docParser : ce qui dans le rectangle rouge est titre détecté

(Figure 4.9) , les deux titres (« Distribution or Dividends » et « Redemptions») ont été considérés comme une partie de la première ligne de bloc de texte respectivement.

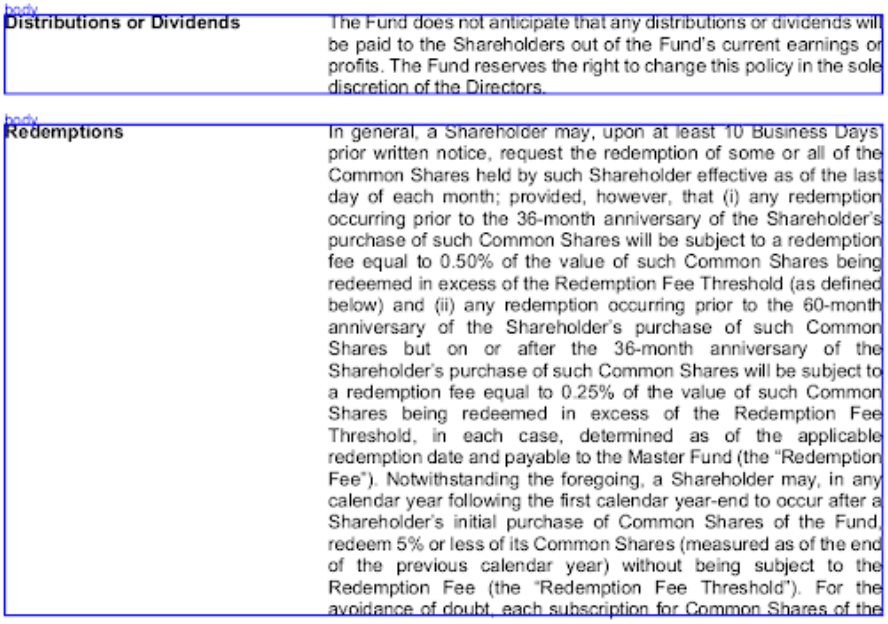


FIGURE 4.9 – La visualisation des résultats de détection de structure par pdfact

Nous pouvons donc observer que le pdfact a détecté l'ensemble du document comme des « body » (blocs de texte) au lieu de deux titres avec des blocs de texte correspondants. Ces deux outils testés ont un problème similaire de délimitations des limites de paragraphes. L'outil docParser, néanmoins, est plus performant sur la détection de titre car avant de détecter les titres, il a vérifié les frontières de bloc de texte et réorganisé ceux qui ne sont pas bons en fonction des coordonnées.

En ce qui concerne les résultats de la détection de bloc de texte correspondant (Tableaux 4.6), les performances des ces deux outils sont similaires. Cela veut dire que dès que les titres sont correctement détectés, la détection de bloc de texte correspondant sera bien détectée, car les blocs de texte correspondantes sont entre deux titres.

Cela pourrait également expliquer la raison pour laquelle la précision de pdfact sur la détection de blocs de texte correspondants est inférieure à celle du docParser. Les titres ont été détectés incorrectement comme des bloc de texte par le pdfact, ce qui augmente le nombre de faux positifs.

Outil	Précision	Rappel	F-mesure
docParser	0,87	0,92	**0,89**
pdfact	0,77	0,86	0,85

TABLE 4.6 – Résultat de l'évaluation de la détection de bloc de texte

4.5 Discussion

Pour un certain nombre de raisons, les problèmes de la conversion des fichiers PDF en texte et des formats des caractères ne sont pas faciles à résoudre même avec les outils performants et industriels. Bien que les documents traités soient tous des documents Offering Memorandum, les documents peuvent être construits de différentes manières, ce qui rend impossible la gestion de chaque PDF de la même manière. Certains outils ne peuvent pas détecter certains formats (e.g. les soulignés), ce qui rend plus difficile la détection des formats combinés (e.g. gras et souligné). De plus, la variation dans les formats des titres peut provoquer des erreurs en termes d'ordre, de sens et de division du texte entre les pages et les blocs. Un autre défi est que la mise en page varie fortement, en particulier lorsqu'il s'agit de documents avec des multi-colonnes ou mixtes avec des pages contenant des parties sans colonne et des parties à double colonnes. C'est la raison pour laquelle qu'il n'est pas possible d'identifier facilement les heuristiques qui permettent d'identifier et d'extraire les textes dans les documents PDF en suivant le bon ordre de lecture.

Tableau 4.7 présente un résumé sur les aspects que les outils testés sont capables de détecter.

Outil	Texte	Frontière de paragraphe	Format	Structure
pdftotext	OUI	OUI	NON	NON
pdfplumber	OUI	NON	OUI	NON
FineReader PDF	OUI	OUI	OUI	NON
pdfact	OUI	OUI	OUI	OUI
pdfExtract	OUI	OUI	OUI	NON
docParser	NON	NON	NON	OUI

TABLE 4.7 – Résumé des aspectes détetables par ces 5 outils

Avant d'effectuer les évaluations, nous pouvons observer que tous les outils testés sont capables d'extraire les textes des documents. Néanmoins, l'outil pdfplumber ne respecte pas les frontières des paragraphes en récupérant les textes d'une page comme un ensemble de données. De plus, seul l'outil pdftotext n'est pas capable de détecter les formats (gras, italique ou souligné). En ce qui concerne la détection de structures de titre et bloc de texte correspondant, les outils pdfact et DocPaser en sont capables.

Pendant l'évaluation de l'extraction de texte, nous observons que les cinq outils sont performants du point de vue de la conversion des fichiers en texte même si FineReader PDF a eu des problèmes pour reconnaître certaines lettres spécifiques. La qualité de la couche textuelle détectée par l'OCR peut varier et les résultats optimaux des actions de copie et de recherche ne peuvent pas être garantis. La plupart des outils à part l'outil pdfact n'ont pas extrait les lignes blanches entre deux paragraphes.

En termes de détection de format, l'outil FineReader PDF est assez performant sur 6 possibilités de format dans le corpus OM : il est capable de les détecter avec

une F-mesure plus de 0,87. En revanche, les outils pdfact et pdfExtract ne sont pas capables de détecter les formats soulignés (le souligné uniquement et le format combiné de gras et souligné). Plus précisément, c'est les librairies intégrées (PDFBox et Vintasoft) qui n'ont pas cette fonction de détecter le format souligné. Cela pourrait s'expliquer par le fait que le soulignement n'est pas un attribut d'une police tels que le gras ou l'italique mais plutôt un objet graphique placé sous le texte selon la documentation de la librairie JAVA PDFBox API [2]. Nous pourrions calculer les coordonnées de ce qui ressemble à une ligne, puis vérifier s'il se trouve en dessous d'un texte.

La précision de l'approche heuristique de docParser est meilleure que la précision de l'approche de pdfact. En effet, les problèmes de délimitation de frontières des chaînes issus de l'étape de l'extraction de texte empêchent l'outil dcoParser de correctement détecter les titres. De manière similaire, l'outil pdfact n'était pas capable de détecter les titres en raison d'une mauvaise délimitation sur les frontières des blocs de texte dans les colonnes. De plus, avec des erreurs de détection de format en souligné, ces deux outils ne sont pas capables de détecter les titres qui sont en souligné uniquement ou en gras et soulignés.

En comparant les approches de ces deux outils, les analyses heuristique de docParser ont pris en compte non seulement les informations de formats (gras et italique) mais également les informations qui permettent de distinguer un titre telles que les couleurs du textes, les indentations et les majuscules. En revanche, l'outil pdfact n'a utilisé que les informations de formats. D'autre part, le docParser a corrigé de mauvaises délimitations de frontières de bloc de texte en prenant en compte les coordonnées. Cela permet d'assurer la précision de la segmentation des paragraphes. Après la détection de titres possibles, le pdfact les a comparés avec une base de données d'articles scientifiques afin de mieux identifier si les chaînes détectées sont des titres. L'approche proposée par [Ramakrishnan et al., 2012] a également corrigé et examiné le bon ordre des blocs de texte à la fin de l'obtention des résultats.

Une autre raison qui pourrait expliquer pourquoi l'outil pdfact est moins performant est qu'il est un outil permettant l'extraction des métadonnées des articles scientifiques dont la structure est différente de celle de documents OM. Les titres des articles scientifiques se situent souvent sur une seule ligne alors que les titres de documents OM peuvent être dans des colonnes. Une autre remarque est que notre définition des titres n'est pas identique. Nous avons définit qu'un titre pouvait être au début d'un paragraphe tant qu'il est considéré comme un titre sémantiquement ou hiérarchiquement, alors que la définition d'un titre pour pdfact est un titre général qui est seul sur une ligne soit en gras ou en italique. Par conséquent, lorsqu'il a extrait les textes, il a extrait bloc de texte par bloc de texte en ignorant les structures à multi-colonnes.

Les approches testées dans ce mémoire ont plusieurs limites, notamment concernant la détection de la structure des documents. Les deux outils analysés sont basés sur les analyses heuristiques qui sont peut-être performantes sur les structures prises en compte. L'identification de formes complexes est une tâche difficile et de longue durée dans un système basé sur des analyses heuristiques [Thanaki, 2017]. Par conséquent, il serait intéressant de mener une expérimentation à base d'une autre approche comme l'approche d'apprentissage automatique ou hybride.

2. https ://pdfbox.apache.org/

Conclusion générale

Pour développer un système d'extraction d'informations pour traiter des documents PDF financiers comme le projet Doc Reader, l'une des étapes les plus importantes est de reconnaître le texte et de détecter la structure de ces documents. La reconnaissance des textes telle que la reconnaissance des frontières des mots, de l'ordre de lecture et des limites des paragraphes permet de transformer un document PDF en un format exploitable. Ces titres détectés dans les documents serviront entre autres à trouver la bonne section afin que les utilisateurs puissent extraire les informations qu'ils souhaitent du paragraphe correspondant à ces titres. Ainsi, la qualité des données collectées et la structure correcte détectée sont essentielles pour la suite du traitement.

Le PDF est l'un des formats de fichiers les plus utilisés pour l'échange et la représentation d'informations. Même si à première vue les documents PDF peuvent se ressembler, il existe différentes manières de les créer. Selon le type de création (directement à partir des applications, en appelant l'API d'impression du système d'exploitation ou un logiciel de tiers), un ensemble d'informations différentes est disponible pour un traitement ultérieur après la création. Les principales catégories dans lesquelles les fichiers PDF peuvent être divisés sont les documents PDF natifs, les documents PDF scannés et les documents PDF multi-couches. Alors que les PDF natifs et les documents multi-couches contiennent toujours des informations de haut niveau sous différentes formes, les PDF scannés sont uniquement des images, n'incluant pas de code numérique pour décrire la structure de formatage. De plus, différentes mises en page (colonne unique, colonnes multiples) et orientations (horizontales, verticales) rendent le sujet de l'extraction de texte à partir de documents PDF très difficile.

L'objectif de ce mémoire était d'effectuer une évaluation sur la performance de l'extraction de texte de document PDF OM. Pour bien comparer la performance de chaque étape, notre travail contenait les évaluations de 1) l'extraction de texte de document PDF en texte, 2) la détection de format de caractères et de 3) la détection de la structure de documents. Cinq outils pour l'extraction de document PDF en texte ont été analysés dont deux utilisant la fonctionnalité OCR. L'outil pdfact est celui qui produit le moins d'erreurs au niveau de la reconnaissance de caractère, tandis que l'outil pdfExtract est le plus performant sur la distinction des paragraphes. En ce qui concerne la détection des formats, l'outil FineReader PDF est le plus performant parmis les trois outils testés (pdfact et pdfExtract) avec une F-mesure de plus de 0.87 pour toutes les possibilités de formats analysées. Les outils pdfact et pdfExtract ne sont pas capables de détecter les textes en soulignés uniquement et les textes en gras et soulignés. En ce qui concerne la détection de structure, les deux outils testés sont basés sur des méthodes heuristiques. L'outil docParser est plus performant que l'outil pdfact avec une F-mesure de 0.84 sur la détection de titre et une F-mesure de 0.89 sur la détection de paragraphes associés. Dans l'ensemble, la reconnaissance du texte, du format, et d'autres informations nécessaires à la détection de la structure

sont des éléments décisifs lors du choix de l'outil approprié.

Certes, les évaluations décrites dans ce mémoire s'accompagnent de plusieurs limites. Il serait intéressant de prendre plus de types de documents financiers afin de comparer la performance de l'extraction de texte et d'avoir une étude plus complète des résultats des différents outils. De plus, un système à base d'une approche d'apprentissage automatique pour la détection de structure pourrait être adoptée pour la comparaison.

BIBLIOGRAPHIE

[Alamoudi et al., 2021] Alamoudi, A., Alomari, A., Alwarthan, S., and Rahman, A. (2021). A rule-based information extraction approach for extracting metadata from pdf books. *ICIC Express Letters*, 12:121–132. – Cité page 29.

[Arora et al., 2017] Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*. – Cité page 52.

[Bast and Korzen, 2013] Bast, H. and Korzen, C. (2013). The icecite research paper management system. volume 8181, pages 396–409. – Cité page 29.

[Bast and Korzen, 2017] Bast, H. and Korzen, C. (2017). A benchmark and evaluation for text extraction from pdf. pages 1–10. – Cité pages 24 et 46.

[Bennamoun et al., 2001] Bennamoun, M., Bergmann, N., and Cheung, A. (2001). An arabic optical character recognition system using recognition-based segmentation. *Pattern Recognition*, 34. – Cité page 26.

[Berg et al., 2012] Berg, , Oepen, S., and Read, J. (2012). Towards high-quality text stream extraction from pdf: Technical background to the acl 2012 contributed task. – Cité page 22.

[Bui et al., 2016a] Bui, D., Del Fiol, G., Hurdle, J., and Jonnalagadda, S. (2016a). Extractive text summarization system to aid data extraction from full text in systematic review development. *Journal of Biomedical Informatics*, 64. – Cité page 24.

[Bui et al., 2016b] Bui, D., Del Fiol, G., and Jonnalagadda, S. (2016b). Pdf text classification to leverage information extraction from publication reports. *Journal of biomedical informatics*, 61. – Cité page 24.

[Chaudhuri et al., 2017] Chaudhuri, A., Mandaviya, K., Badelia, P., and Ghosh, S. (2017). Optical character recognition systems for different languages with soft computing. 352. – Cité page 27.

[Damerow et al., 2017] Damerow, J., Peirson, E., and Laubichler, M. (2017). The giles ecosystem – storage, text extraction, and ocr of documents. *Journal of Open Research Software*, 5. – Cité page 24.

[Doucet et al., 2013] Doucet, A., Kazai, G., Colutto, S., and Muhlberger, G. (2013). Overview of the icdar 2013 competition on book structure extraction. pages 1438–1443. – Cité page 29.

[EL-Haj et al., 2019a] EL-Haj, M., Alves, P., Rayson, P., Walker, M., and Young, S. (2019a). Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as pdf files. *Accounting and Business Research*, 50:1–29. – Cité page 29.

[EL-Haj et al., 2019b] EL-Haj, M., Rayson, P., Alves, P., Herrero-Zorita, C., and Young, S. (2019b). *Multilingual Financial Narrative Processing: Analyzing Annual Reports in English, Spanish, and Portuguese: Challenges, Models, and Approaches*, pages 441–463. – Cité page 29.

[EL-Haj et al., 2014] EL-Haj, M., Rayson, P., Young, S., and Walker, M. (2014). Detecting document structure in a very large corpus of uk financial reports. – Cité page 29.

[Forgy, 1990] Forgy, C. (1990). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Expert Systems*, pages 324–341. – Cité page 30.

[Guedes and Silva, 2021] Guedes, G. and Silva, A. (2021). *Supervised Learning Approach for Section Title Detection in PDF Scientific Articles*, pages 44–54. – Cité page 29.

[Hayes, 2020] Hayes, A. (2020). Offering memorandum. – Cité page 35.

[I. Y. Korneev et al., 2015] I. Y. Korneev, S. G. Popov, A. S. M., N. Kolodkina, S., and Read, J. (2015). Retention of content in converted documents. – Cité page 23.

[Islam et al., 2016] Islam, N., Islam, Z., and Noor, N. (2016). A survey on optical character recognition system. *ITB Journal of Information and Communication Technology*. – Cité page 26.

[Lovegrove and Brailsford, 1995] Lovegrove, W. and Brailsford, D. (1995). Document analysis of pdf files: methods, results and implications. 8. – Cité page 12.

[Masum et al., 2018] Masum, M., Kosaraju, S. C., Bayramoglu, T., Modgil, G., and Kang, M. (2018). Automatic knowledge extraction from ocr documents using hierarchical document analysis. pages 189–194. – Cité page 26.

[McConnaughey et al., 2017] McConnaughey, L., Dai, J., and Bamman, D. (2017). The labeled segmentation of printed books. pages 737–747. – Cité page 29.

[Needleman, 1970] Needleman, S. (1970). Wunsch. a general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 3. – Cité page 30.

[Orion, 2007] Orion, E. (2007). Pdf 1.7 is approved as iso 32000. *The Inquirer*. – Cité page 21.

[Patel et al., 2012] Patel, C., Patel, A., and Patel, D. (2012). Optical character recognition by open source ocr tool tesseract: A case study. *International Journal of Computer Applications*, 55:50–56. – Cité pages 26 et 27.

[Ramakrishnan et al., 2012] Ramakrishnan, C., Patnia, A., Hovy, E., and Burns, G. (2012). Layout-aware text extraction from full-text pdf of scientific articles. *Source code for biology and medicine*, 7:7. – Cité pages 30 et 56.

[Rigamonti et al., 2004] Rigamonti, M., Hadjar, K., Lalanne, D., and Ingold, R. (2004). Xed : un outil pour l'extraction et l'analyse de documents pdf. – Cité pages 12 et 28.

[Smith et al., 1981] Smith, T. F., Waterman, M. S., et al. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197. – Cité page 47.

[Strecker et al., 2009] Strecker, T., Van Beusekom, J., Albayrak, S., and Breuel, T. M. (2009). Automated ground truth data generation for newspaper document images. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1275–1279. IEEE. – Cité page 37.

[Teufel, 2010] Teufel, S. (2010). The structure of scientific articles: Applications to citation indexing and summarization. *Bibliovault OAI Repository, the University of Chicago Press*. – Cité page 29.

[Thanaki, 2017] Thanaki, J. (2017). *Python natural language processing*. Packt Publishing Ltd. – Cité page 56.

[van der Knijff, 2009a] van der Knijff, J. (2009a). Adobe portable document format. *Inventory of long-term preservation risks, v0*, 2:20–56. – Cité pages 21 et 22.

[van der Knijff, 2009b] van der Knijff, J. (2009b). Adobe portable document format. *Inventory of long-term preservation risks, v0*, 2:20–56. – Cité page 25.

[Warnock, 1991] Warnock, J. (1991). The camelot project. *PDF). PlanetPDF." This document describes the base technology and ideas behind the project named "Camelot." This project's goal is to solve a fundamental problem [...] there is no universal way to communicate and view... printed information electronically*. – Cité page 21.

[Whitington, 2011] Whitington, J. (2011). *PDF explained. In (chap. I Introduction)*. – Cité page 22.

APPENDIX A

ANNEXE

A.1 Liste des champs extractibles dans l'application DOC Reader (extrait)

Nom du champ	**Type du champ** (dans le cadre d'un projet X (anonyme))
Auditor Name	l'information attendue correspond aux entités prédéfinies
Custodian Bank Name	l'information attendue correspond aux entités prédéfinies
Domicile Of ManCo	l'information attendue correspond aux entités prédéfinies
Fiscal Year End	l'information attendue correspond à une phrase
Legal Fund Name Only	l'information attendue correspond aux entités prédéfinies
Fund Manager Name	l'information attendue correspond aux entités
Performance Fee	l'information attendue correspond à une section entière
Investment Objective	l'information attendue correspond à une section entière.
Investment Management Fee	l'information attendue correspond à une section entière.
Minimal Initial Subscription In Shares	l'information attendue correspond aux phrases
Minimal Initial Redemption In Shares	l'information attendue correspond à une section entière.
Fees, Costs and expenses	l'information attendue correspond à une section entière.

Il n'y a pas de nombre limite sur les champs détectables dans cette application. Le type du champ est également modulable en fonction des besoins des utilisateurs et des modèles.

A.2 Format de fichiers JSON des outils et de référence pour l'évaluation de la détection de format de caractère.

1. Résultats par FineReader PDF convertis en format JSON :


```
% \begin{minted}{json}
{
          "sequence": "SUMMARY OF PRINCIPAL TERMS",
          "sequence_id": 39,
          "tokens": [
            {
              "bold": 1,
              "italic": 0,
              "underline": 0,
              "token": "SUMMARY"
            },
            {
              "bold": 1,
              "italic": 0,
              "underline": 0,
              "token": "OF"
            },
            {
              "bold": 1,
              "italic": 0,
              "underline": 0,
              "token": "PRINCIPAL"
            },
            {
              "bold": 1,
              "italic": 0,
              "underline": 0,
              "token": "TERMS"
            }
          ]
        }
```


2. Résultats par pdfExtract convertis en format JSON :


```
% \begin{minted}{json}
{
          "page_number": 7,
          "lines": [
            {
              "tokens": [
                {
                  "token": "SUMMARY",
                  "bold": 1,
                  "italic": 0,
```

```
                    "underline": 0
                },
                {
                    "token": "OF",
                    "bold": 1,
                    "italic": 0,
                    "underline": 0
                },
                {
                    "token": "PRINCIPAL",
                    "bold": 1,
                    "italic": 0,
                    "underline": 0
                },
                {
                    "token": "TERMS",
                    "bold": 1,
                    "italic": 0,
                    "underline": 0
                }
            ],
            "line_text": "SUMMARY OF PRINCIPAL TERMS"
        }
% \end{minted}
```

3. Résultats par pdfact convertis en format JSON :

```
{
  "page_number": 7,
  "line_id": 218,
  "tokens": [
    {
      "token": "SUMMARY",
      "bold": 1,
      "italic": 0,
      "underline": 0
    },
    {
      "token": "OF",
      "bold": 1,
      "italic": 0,
      "underline": 0
    },
    {
      "token": "PRINCIPAL",
      "bold": 1,
      "italic": 0,
      "underline": 0
    },
    {
```

```
      "token": "TERMS",
      "bold": 1,
      "italic": 0,
      "underline": 0
    }
  ],
  "line_text": "SUMMARY OF PRINCIPAL TERMS "
}
```

4. Corpus de référence en format JSON :

```
{
          "page_number": "7",
          "title": "SUMMARY OF PRINCIPAL TERMS",
          "title_id": "17",
          "tokens": [
            {
              "token": "SUMMARY",
              "bold": 1,
              "italic": 0,
              "underline": 0
            },
            {
              "token": "OF",
              "bold": 1,
              "italic": 0,
              "underline": 0
            },
            {
              "token": "PRINCIPAL",
              "bold": 1,
              "italic": 0,
              "underline": 0
            },
            {
              "token": "TERMS",
              "bold": 1,
              "italic": 0,
              "underline": 0
            }
          ]
        }
```

A.3 Résultats d'un titre en gras et en souligné de pdfExtract et FineReader PDF

Résultats de pdfExtract


```
{
       "tokens": [
             {
             "token": "Nature",
              "bold": 0,
              "italic": 0,
              "underline": 0
                },
                {
               "token": "of"
               "bold": 0,
               "italic": 0,
               "underline": 0
                },
                {
               "token":
"Investments",
               "bold": 0,
               "italic": 0,
               "underline": 0
                }
              ],
              "line_text":
"Nature of Investments "
            }
```


Résultats de FineReader PDF


```
{
          "sequence": "Nature of
Investments",
          "sequence_id": 64,
          "tokens": [
            {
              "bold": 1,
              "italic": 0,
              "underline": 1,
              "token": "Nature"
            },
            {
              "bold": 1,
              "italic": 0,
              "underline": 1,
              "token": "of"
            },
            {
              "bold": 1,
              "italic": 0,
              "underline": 1,
              "token":
"Investments"
            }
          ]
        }
```


Cela montre que l'outil FineReader PDF est capable de détecter les chaînes à la fois en gras et en souligné alors que l'outil pdfExtract ne l'est pas capable.