République Algérienne Démocratique et Populaire

Ministére de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaia

Faculté des Sciences Exactes

Département de Recherche Opérationnelle

Apprentissage profond et automatisation de l'extraction d'information des Cvs (Cas Tech'Instinct)

Par Hamami Athmane Ryane

Mémoire de fin de cycle En vue d'obtention du diplôme de Master en Mathématiques appliquées

Spécialité Sciences des donnees et Aide a la decision

Présentée et soutenue publiquement le 04/07/2024

Devant un jury composé de :

Mme. F. AOUDIA	UAMB - Bejaia	Encadrante
Mlle. Z. AOUDIA	UAMB - Bejaia	Encadrante
M. Y. BELATTAF	CEO Tech'Instinct	Invité
Mme. S. BOULERKHAR	UAMB - Bejaia	Présidente
Mme. S. AMROUN	UAMB - Bejaia	Examinateur
M. M'HAMDI	UAMB - Bejaia	Examinateur

Mon frère Ma soeur, écoute :

La loi de l'attraction n'existe pas. Dire que je suis riche un million de fois ne fera pas apparaître l'argent par magie. Ce n'est pas comme ça que ça marche.

L'Islam a toujours eu les réponses, laisse-moi t'expliquer. Dans un hadith qudsi, Allah le Tout-Puissant dit : "Je suis tel que mon serviteur pense que je suis..." Si tu penses qu'Allah te donnera de la richesse, tu obtiendras de la richesse. Si tu penses qu'Allah va te mettre dans le manque, c'est ce que tu obtiendras. Il ne s'agit pas de ce que tu penses de toi-même, mais de ce que tu penses d'Allah.

Alors, comment peux-tu exploiter ce pouvoir ? Active ta conscience divine, vis dans un état de taqwa, sachant qu'Allah te surveille. Évite le haram, cela te protège de la négativité et te garde calme et concentré.

Demande-toi ce que tu veux vraiment. L'esprit est souvent limité et programmé par les influences sociétales. Au lieu d'objectifs arbitraires comme 10 000 euros par mois, demande à ton cœur ce que tu désires vraiment.

N'idolâtre pas tes objectifs, aucun objectif n'est trop grand pour Allah. Comprendre la promesse d'Allah d'abondance dans l'au-delà rend les objectifs terrestres insignifiants.

Vis comme si c'était déjà arrivé. Crois qu'Allah peut répondre à tes demandes, agis et pense comme si tes objectifs avaient été atteints. Ton subconscient te guidera automatiquement vers ta réussite.

Et enfin, abandonne-toi et ne résiste pas. Accepte tout ce qui arrive comme faisant partie du plan d'Allah. Ne résiste pas aux changements ou aux revers, considère-les comme des étapes vers ton objectif final. Agis avec confiance et persévérance. Ne complique pas les choses, fais confiance à Allah et fais du'a pour rendre les choses plus claires.

Ryane Hamami

Remerciements

Je tiens à prendre cette page pour exprimer ma reconnaissance envers toute personne ayant contribué à la réalisation de ce projet de fin de cycle, que ce soit de loin ou de près.

À mes deux encadrantes, Mlle Z. Aoudia et Mme F. Aoudia, je vous adresse toute ma gratitude pour votre accompagnement tout au long de ce travail. Vos conseils précieux et votre soutien m'ont permis de progresser continuellement. Votre expertise et votre bienveillance ont été des éléments clés dans la realisation de ce memoire.

Je tiens à adresser mes sincères remerciements à mon superviseur en entreprise, M. Youcef Belattaf. Je suis reconnaissant pour l'opportunité qui m'a été offerte et pour sa confiance placée en moi pour mener à bien le projet de mon stage. Ce projet, empreint de Data Science et d'IA, ainsi que l'aspect collaboratif sur les différents sujets liés, ont créé un environnement où l'innovation est valorisée. Merci pour votre soutien inébranlable.

À mes confrères, Chafaa Kherib, Samy Outamzabet, Mehani Sabri et Djoudi Mansouri, je vous remercie pour votre bienveillance en m'intégrant aussi bien dans l'environnement de travail lors de mon stage et pour vos conseils avisés. Votre expertise et votre soutien ont été essentiels pour mon développement professionnel.

À ma famille, en particulier à ma mère, je t'en suis infiniment reconnaissant pour ton soutien constant. Chaque seconde de ma vie, tu as été derrière moi, et sans tous tes sacrifices, rien de tout cela ne serait possible. Ton amour et tes encouragements sont ma principale motivation pour atteindre mes objectifs.

À mes amies, Mehdi et Salas, dont le soutien et la camaraderie ont été inestimables durant mon parcours universitaire. Mes remerciements vont naturellement aussi à Céline qui m'apporte du courage et de la joie.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

Abréviation	Signification
CV	Curriculum Vitae
NLP	Traitement du Langage Naturel (Natural Language Processing)
GPU	Unité de Traitement Graphique (Graphics Processing Unit)
CNN	Convolutional Neural Networks
OCR	Optical Character Recognition (Reconnaissance optique de caractères)
ML	Apprentissage Automatique (Machine Learning)
DL	Apprentissage profond (Deep Learning)
GPU	Unité de Traitement Graphique
NER	Named Entity Recognition (reconnaissance d'entité nommée)
LLM	Large language Model (Grand modèle de langage)
RNN	Recurrent neural network (réseaux de neurones récurrents)
LSTM	Long short term memory (réseaux de longue mémoire à court terme)
BERT	Bidirectional Encoder Representations from Transformers

Introduction générale

Dans un monde en constante évolution numérique, où la digitalisation s'intensifie et l'automatisation des processus devient cruciale, l'extraction d'informations à partir de documents non structurés, en particulier les curriculum vitae (CV), représente un défi technique majeur et une opportunité significative. Comme le soulignent Guo et al. "l'extraction automatique d'informations à partir de CV est devenue un enjeu crucial pour les entreprises cherchant à optimiser leurs processus de recrutement et de gestion des talents" [1]. Ce mémoire se concentre sur le développement et l'implémentation d'une approche innovante et automatisée pour l'extraction d'informations des CV, en exploitant des techniques avancées d'apprentissage profond et de traitement du langage naturel (NLP).

Notre objectif principal est de créer un système robuste et efficace capable d'extraire avec précision des informations pertinentes à partir de CV sous divers formats. Pour atteindre cet objectif, nous avons conçu une approche multidisciplinaire qui combine des techniques de vision par ordinateur pour l'analyse structurelle des documents, La reconnaissance optique de caractères (OCR) pour l'extraction de texte et des méthodes avancées de traitement du langage naturel pour l'analyse et la structuration des informations.

La première étape de notre approche consiste à analyser la structure des CV pour comprendre l'organisation des informations qu'ils contiennent. Cette étape est cruciale pour faciliter l'extraction ciblée des données pertinentes. Nous utilisons des techniques de détection d'objets, notamment les modèles Faster R-CNN et Mask R-CNN, pour identifier visuellement les différents éléments structurels des CV, tels que les sections de texte.

On second plan on s'est attarder sur une analyse des techniques de reconnaissance optique de caractères (OCR). L'OCR joue un rôle central dans notre système, permettant l'extraction de texte brut à partir de documents fournis sous divers formats (images, PDF, etc.). Comme l'expliquent Zhang et al, "l'utilisation de techniques d'OCR avancées permet d'améliorer significativement la précision de l'extraction de texte à partir de documents complexes tels que les CV" [2]. Nous avons évalué et comparé les performances de plusieurs bibliothèques open-source d'OCR, notamment EasyOCR, PaddleOCR et Tesseract, pour leur robustesse et leur capacité à traiter efficacement divers formats de documents.

Une fois le texte brut extrait, nous appliquons des techniques avancées de NLP pour comprendre et structurer les informations. Nous utilisons des méthodes de reconnaissance d'entités nommées (NER) pour identifier des éléments clés tels que les noms, les entreprises, les diplômes, etc. Notre approche s'appuie sur des modèles pré-entraînés de Transformers de Hugging Face, que nous affinons pour notre tâche spécifique. Pour améliorer les perfor-

mances des modèles NER, nous avons procédé au fine-tuning de modèles pré-entraînés sur des données de CV annotées. Nous avons notamment utilisé CamemBERT, une variante de BERT optimisée pour le français, ainsi que Mistral 7B, un modèle LLM conçu pour le traitement du texte en français. Selon Dupont et al, "l'utilisation de modèles pré-entraînés spécifiques à la langue, tels que CamemBERT pour le français, combinée à un fine-tuning approprié, permet d'obtenir des performances significativement meilleures dans les tâches d'extraction d'informations à partir de documents en français" [3].

Nous avons intégré ces différentes composantes dans un workflow global qui combine l'analyse de structure, l'OCR, le NLP et l'apprentissage profond. Ce mémoire détaille les étapes de notre démarche, les choix technologiques effectués, ainsi que les résultats obtenus. Nous présentons également une analyse critique de notre approche, identifiant les axes d'amélioration pour optimiser davantage notre système.

Ce travail démontre que l'utilisation combinée des techniques d'apprentissage profond, de reconnaissance optique de caractères et de traitement du langage naturel permet de créer un système performant pour l'extraction automatique d'informations à partir de CV. Les résultats obtenus montrent des avancées significatives dans la précision et l'efficacité de l'extraction d'informations, tout en ouvrant la voie à de futures améliorations et applications potentielles dans le domaine des ressources humaines et au-delà.

Chapitre 1

Fondements du Machine Learning et du Deep Learning

1.1 Introduction

L'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning) ont connu une évolution spectaculaire au cours des dernières décennies. Depuis les premiers travaux des années 1950-1960 sur la régression linéaire et la classification, en passant par le développement d'algorithmes d'apprentissage plus sophistiqués comme les réseaux de neurones artificiels dans les années 1980-1990, ce champ de recherche n'a cessé de progresser.

Parmi les articles de recherche les plus marquants, on peut citer les travaux pionniers de Yann LeCun, Yoshua Bengio et Geoffrey Hinton. En 1998, ont publié un article fondateur sur les réseaux de neurones convolutifs appliqués à la reconnaissance d'images manuscrites. Quelques années plus tard, Bengio et al. ont proposé une méthode innovante pour l'entraînement de réseaux de neurones profonds.

Aujourd'hui, le machine learning et le deep learning sont omniprésents, mais soulèvent aussi des défis éthiques importants à relever, comme les biais algorithmiques et la protection de la vie privée [4]. Ce domaine en constante évolution continue de façonner notre monde, et il est essentiel de comprendre ses fondements pour appréhender les opportunités et les responsabilités qu'il engendre.

1.2 Conceptualisation du Machine Learning

1.2.1 Définition du Machine Learning

Le Machine Learning, ou apprentissage automatique en français, est une branche de l'intelligence artificielle qui permet aux systèmes informatiques d'apprendre et de s'améliorer

à partir de l'expérience sans être explicitement programmés. Comme le soulignent Alpaydin et al. (2016), "le Machine Learning vise à construire des systèmes qui peuvent apprendre et s'améliorer par eux-mêmes à partir de données, sans avoir besoin d'être programmés de manière explicite".[5] Contrairement aux programmes traditionnels qui suivent des instructions strictes, les algorithmes de Machine Learning sont capables de détecter des modèles dans les données et d'ajuster leur comportement en conséquence.

L'essence du Machine Learning réside dans sa capacité à généraliser à partir des données observées pour faire des prédictions ou prendre des décisions sur de nouvelles données non vues auparavant. Selon Hastie et al. (2009), "l'objectif principal du Machine Learning est de développer des techniques qui permettent aux ordinateurs d'apprendre à partir des données, plutôt que de s'appuyer uniquement sur des règles programmées explicitement".[6] En d'autres termes, il s'agit d'apprendre à partir de l'expérience passée pour effectuer des tâches spécifiques dans le futur.

Cette discipline repose sur un ensemble de techniques et d'algorithmes, allant des méthodes classiques telles que la régression linéaire et les arbres de décision aux modèles plus avancés tels que les réseaux de neurones profonds. L'objectif ultime du Machine Learning est de créer des systèmes capables d'imiter et d'améliorer les capacités humaines d'apprentissage et de prise de décision.

1.2.2 Historique et évolution du Machine Learning

L'histoire du Machine Learning remonte aux débuts de l'informatique et de l'intelligence artificielle. Ses fondements théoriques remontent aux travaux pionniers de chercheurs tels que Alan Turing et Claude Shannon dans les années 1940 et 1950. Cependant, ce n'est qu'au cours des dernières décennies que le domaine a connu une explosion d'intérêt et de progrès significatifs.

Dans les années 1950 et 1960, les premiers travaux sur l'apprentissage automatique ont été largement axés sur les modèles de régression et les premiers algorithmes d'apprentissage supervisé. Les années 1980 et 1990 ont été marquées par des avancées dans les réseaux neuronaux, bien que leur utilisation ait été limitée en raison de contraintes matérielles et de performances insuffisantes.

L'émergence d'Internet et l'explosion des données numériques dans les années 2000 ont ouvert de nouvelles opportunités pour le Machine Learning. L'essor des grands ensembles de données (big data) et des capacités de calcul distribué ont permis le développement de modèles plus complexes et l'application de techniques telles que le Deep Learning.

Aujourd'hui, le Machine Learning est omniprésent dans de nombreux aspects de notre vie quotidienne, des moteurs de recherche et des recommandations de produits en ligne aux véhicules autonomes et à la médecine personnalisée. Son évolution continue de façon exponentielle, portée par des avancées dans l'algorithmique, le matériel informatique et la disponibilité croissante des données.

1.2.3 Applications et domaines d'utilisation du Machine Learning

Le Machine Learning (apprentissage automatique) trouve de vastes applications dans divers domaines et industries, révolutionnant notre approche des problèmes et des prises de décisions. Voici quelques exemples concrets des principaux domaines où le Machine Learning est largement utilisé :

Finance : Dans le domaine financier, le Machine Learning est utilisé pour prévoir les mouvements des marchés financiers en analysant des données historiques et en identifiant des tendances et des schémas. Cela aide les investisseurs à prendre des décisions éclairées. De plus, le Machine Learning est utilisé pour détecter les fraudes financières en analysant les modèles de dépenses et en identifiant les transactions suspectes.

Commerce électronique : Le Machine Learning joue un rôle essentiel dans les recommandations de produits personnalisées sur les sites de commerce électronique. Les algorithmes de Machine Learning analysent les préférences des utilisateurs, leur historique d'achat et les données démographiques pour recommander des produits pertinents. Par exemple, si un utilisateur achète un téléphone portable, le système peut recommander des accessoires compatibles tels que des étuis de protection ou des écouteurs sans fil.

Technologie : Le Machine Learning est au cœur de nombreuses technologies que nous utilisons quotidiennement. Par exemple, la reconnaissance vocale permet aux assistants virtuels tels que Siri ou Alexa de comprendre et de répondre à nos commandes vocales. La traduction automatique utilise également des techniques de Machine Learning pour traduire des textes d'une langue à une autre avec précision. De plus, le traitement du langage naturel permet aux systèmes de comprendre et d'interpréter le langage humain, ce qui est utilisé dans les chatbots et les systèmes de recherche intelligents.

Transport : Le Machine Learning est appliqué à la navigation autonome, où les véhicules utilisent des algorithmes de Machine Learning pour détecter les objets, comprendre les signaux routiers et prendre des décisions en temps réel. L'optimisation des itinéraires utilise également le Machine Learning pour analyser les données de trafic en temps réel et proposer les itinéraires les plus rapides et les plus efficaces. De plus, le Machine Learning est utilisé pour la maintenance prédictive des véhicules, en analysant les données des capteurs pour prédire les pannes et planifier les opérations de maintenance.

Industrie : Dans le domaine de l'industrie, le Machine Learning est utilisé pour la fabrication intelligente, où les machines apprennent à optimiser les processus de production en analysant les données en temps réel et en adaptant les paramètres en conséquence. La surveillance des équipements utilise également le Machine Learning pour détecter les défaillances potentielles et prévenir les pannes avant qu'elles ne se produisent. De plus, le contrôle de qualité des produits peut être amélioré en utilisant des algorithmes de Machine Learning pour détecter les défauts ou les anomalies dans les produits fabriqués.

Marketing et Vente : Le Machine Learning est utilisé pour analyser les données des clients et comprendre leurs préférences, leurs comportements d'achat et leurs habitudes. Ces informations sont ensuite utilisées pour segmenter le marché et personnaliser les offres et les

publicités en fonction des besoins spécifiques de chaque segment. De plus, l'automatisation du marketing utilise le Machine Learning pour automatiser les campagnes publicitaires, les e-mails personnalisés et les recommandations de produits, ce qui permet d'améliorer l'efficacité et la pertinence des efforts marketing.

1.3 Acquisition et Préparation des Données

1.3.1 Types de données (structurées, non structurées, semi-structurées)

Les données peuvent être classées en trois principaux types : structurées, non structurées et semi-structurées. Les données structurées sont organisées dans un format tabulaire avec des lignes et des colonnes, telles que les données stockées dans une base de données relationnelle. Par exemple, une base de données clients avec des colonnes telles que "Nom", "Prénom", "Âge" et "Adresse" est un exemple de données structurées.

Comme le souligne un article de la revue "Data Mining and Knowledge Discovery", "Les données structurées, telles que celles stockées dans des bases de données relationnelles, permettent un accès et une analyse rapides, mais ne représentent qu'une partie des informations disponibles. Les données non structurées, comme les documents textuels ou les fichiers multimédia, contiennent souvent des informations précieuses mais sont plus difficiles à exploiter."[7].

Les données non structurées, en revanche, ne suivent pas un format tabulaire et ne sont pas organisées de manière prévisible. Elles peuvent inclure des données telles que des documents texte, des images, des fichiers audio et vidéo. Par exemple, des commentaires sur les réseaux sociaux ou des fichiers audio de conversations téléphoniques sont des exemples de données non structurées.

Bien que les données structurées restent la base de nombreuses applications, "l'explosion des données non structurées, notamment issues des réseaux sociaux et des capteurs connectés, offre de nouvelles opportunités d'analyse et de prise de décision", comme le souligne un article de la revue scientifique "Big Data Research".[8].

Les données semi-structurées se situent quelque part entre les deux. Elles ont une certaine organisation, mais ne sont pas aussi rigides que les données structurées. Un exemple courant de données semi-structurées est le format XML, où les données sont organisées avec des balises mais ne suivent pas un schéma fixe comme dans une base de données relationnelle. Selon un article de la revue "Information Systems", "Les données semi-structurées, telles que les fichiers XML ou JSON, combinent les avantages de la structure organisée des données avec la flexibilité des données non formatées. Elles permettent de capturer des informations complexes tout en facilitant leur traitement informatique."[9].

1.3.2 Processus de collecte et de préparation des données

Le processus de collecte et de préparation des données est une étape cruciale dans tout projet d'analyse de données. Il comprend plusieurs étapes, notamment la collecte des données, le nettoyage, la transformation et la validation. La collecte des données peut se faire à partir de différentes sources telles que des bases de données, des fichiers CSV, des API web ou des capteurs.

Une fois les données collectées, elles doivent être nettoyées pour éliminer les erreurs, les valeurs manquantes et les incohérences. Par exemple, si une colonne contient des valeurs manquantes, vous pouvez les remplir en utilisant des méthodes telles que l'imputation de la moyenne ou de la médiane. Ensuite, les données sont souvent transformées et normalisées pour les préparer à l'analyse. Par exemple, vous pouvez appliquer une transformation logarithmique à des données fortement asymétriques pour les rendre plus symétriques et plus adaptées à certains algorithmes.

Enfin, les données sont validées pour garantir leur qualité et leur intégrité. Cela peut inclure des vérifications de cohérence et des tests de conformité aux contraintes de données. Une fois que les données ont été collectées, nettoyées et préparées, elles sont prêtes à être utilisées pour l'analyse et la modélisation.

1.3.3 Techniques de nettoyage, normalisation et transformation des données

Le nettoyage, la normalisation et la transformation des données sont des étapes essentielles du processus de prétraitement des données. Le nettoyage des données vise à éliminer les erreurs et les valeurs aberrantes qui pourraient fausser les résultats de l'analyse. Par exemple, vous pouvez supprimer les doublons, remplacer les valeurs manquantes ou corriger les erreurs de saisie.

La normalisation des données consiste à mettre les données à une échelle commune pour faciliter la comparaison et l'analyse. Par exemple, si vous avez des variables avec des plages de valeurs très différentes, telles que l'âge et le revenu, vous pouvez les normaliser pour les ramener à une échelle commune, comme une plage de 0 à 1.

La transformation des données implique de modifier la structure ou la forme des données pour les rendre plus adaptées à l'analyse. Par exemple, vous pouvez créer de nouvelles variables en combinant des variables existantes, ou appliquer des transformations mathématiques pour rendre les données plus conformes aux hypothèses des modèles statistiques.

En utilisant ces techniques de nettoyage, de normalisation et de transformation des données, les analystes peuvent s'assurer que leurs données sont de haute qualité et prêtes à être utilisées pour l'analyse et la modélisation.

1.4 Algorithmes Traditionnels de Machine Learning

1.4.1 Régression linéaire et logistique

La régression linéaire et logistique sont des algorithmes fondamentaux en statistiques et en apprentissage automatique. La régression linéaire est utilisée pour modéliser la relation entre une variable dépendante continue et une ou plusieurs variables indépendantes, tandis que la régression logistique est utilisée pour modéliser des variables binaires.

Par exemple, dans le cas de la régression linéaire, supposons que nous voulions prédire le prix d'une maison en fonction de sa taille en pieds carrés. Nous pourrions utiliser un modèle de régression linéaire où la taille de la maison est la variable indépendante et le prix de la maison est la variable dépendante. "La régression linéaire multiple est l'une des méthodes les plus puissantes et les plus polyvalentes de l'analyse statistique. Elle permet de quantifier la relation entre une variable dépendante continue et plusieurs variables indépendantes."[10]

Pour la régression logistique, considérons un exemple où nous voulons prédire si un e-mail est spam ou non en fonction de ses caractéristiques telles que le nombre de mots-clés suspects et la présence de pièces jointes. Dans ce cas, la variable dépendante serait binaire (spam ou non-spam), et le modèle de régression logistique apprendrait à modéliser la probabilité qu'un e-mail soit spam en fonction de ses caractéristiques. "La régression logistique est une technique puissante pour modéliser des variables dépendantes binaires ou dichotomiques. Elle permet d'estimer la probabilité qu'un événement se produise en fonction de variables explicatives."[11]

1.4.2 Arbres de décision et forêts aléatoires

Les arbres de décision sont des modèles d'apprentissage supervisé utilisés pour la classification et la régression. Ils fonctionnent en partitionnant de manière récursive l'espace des caractéristiques en sous-ensembles homogènes, basés sur les valeurs des caractéristiques. Les forêts aléatoires sont des ensembles d'arbres de décision, où plusieurs arbres sont formés sur des sous-ensembles aléatoires des données d'entraînement et les résultats sont agrégés pour obtenir des prédictions plus robustes.

Par exemple, dans le cas des arbres de décision, considérons un ensemble de données contenant des informations sur des clients, y compris leur âge, leur revenu et leur historique d'achat, et nous voulons prédire s'ils achèteront ou non un produit spécifique. Un arbre de décision pourrait être utilisé pour partitionner l'ensemble de données en sous-groupes basés sur des caractéristiques telles que l'âge et le revenu, pour prédire les achats. "Les méthodes d'apprentissage automatique comme les arbres de décision sont des outils puissants pour analyser des données complexes et trouver des relations non linéaires entre les variables."[10]

Les forêts aléatoires étendent cette idée en construisant plusieurs arbres de décision indépendants et en agrégeant leurs prédictions. Cela permet d'atténuer le surajustement

(overfitting) et d'améliorer la généralisation du modèle, en faisant des forêts aléatoires une méthode puissante pour une variété de tâches de classification et de régression. "Les forêts aléatoires combinent les prédictions de multiples arbres de décision pour obtenir des prédictions plus robustes et précises que chaque arbre pris individuellement."[10]

1.4.3 Algorithme des k plus proches voisins

L'algorithme des k plus proches voisins, sont des algorithmes d'apprentissage supervisé utilisés pour la classification et la régression. Ils fonctionnent en cherchant les points de données similaires (voisins) dans l'espace des caractéristiques et en utilisant leur étiquette ou leur valeur pour prédire la cible d'un nouveau point de données.

Par exemple, dans le cas de la classification k-NN, supposons que nous avons un ensemble de données contenant des fleurs avec des caractéristiques telles que la longueur et la largeur des pétales et des sépales, ainsi que leur espèce (par exemple, iris setosa, iris versicolor, iris virginica). Pour prédire l'espèce d'une nouvelle fleur, nous pourrions trouver les k fleurs les plus similaires dans l'ensemble de données et utiliser la classe majoritaire parmi ces voisins comme prédiction. "Les méthodes de voisinage comme k-NN sont des techniques simples mais efficaces pour la classification et la régression, en se basant sur la proximité des points de données dans l'espace des caractéristiques."[10]

Les méthodes de voisinage sont simples à comprendre et à mettre en œuvre, mais elles peuvent être sensibles à la dimensionnalité et nécessitent souvent une normalisation des caractéristiques pour des performances optimales. Malgré cela, ils sont largement utilisés pour leur simplicité et leur efficacité dans de nombreuses applications de classification et de régression.

1.5 Deep Learning

Le Deep Learning, ou apprentissage profond, est une branche avancée du Machine Learning qui s'appuie sur des réseaux de neurones artificiels pour modéliser et comprendre des données complexes. Contrairement au Machine Learning traditionnel, qui se base souvent sur des algorithmes relativement simples comme la régression linéaire ou les arbres de décision, le Deep Learning utilise des architectures neuronales profondes avec de multiples couches de neurones pour apprendre des représentations hiérarchiques et de plus en plus abstraites des données.

"Le deep learning utilise des réseaux neuronaux profonds composés de plusieurs couches de neurones interconnectés, ce qui permet l'extraction automatique de caractéristiques et l'apprentissage par représentation."[12] Cette capacité à apprendre des représentations adaptées à partir des données elles-mêmes, plutôt que de devoir les définir à la main, est l'un des principaux avantages du Deep Learning par rapport aux approches plus traditionnelles.

Contrairement au machine learning traditionnel, le Deep Learning ne nécessite pas de conception manuelle des caractéristiques. Les réseaux de neurones apprennent eux-mêmes

les représentations les plus appropriées à partir des données. Cela permet au Deep Learning de résoudre des problèmes complexes de manière plus performante, en identifiant automatiquement les motifs et les abstractions les plus pertinents dans les données.

Le Deep Learning se distingue du machine learning traditionnel par sa capacité à apprendre des représentations hiérarchiques complexes des données, ce qui permet de résoudre des problèmes plus difficiles de manière plus performante. Cette aptitude à extraire des représentations de plus en plus élaborées fait du Deep Learning un outil particulièrement puissant pour la résolution de problèmes complexes dans de nombreux domaines, de la vision par ordinateur à la reconnaissance vocale en passant par le traitement du langage naturel.

1.5.1 Architecture et fonctionnement des réseaux de neurones artificiels

Les réseaux de neurones artificiels (RNA) sont des modèles computationnels s'inspirant du fonctionnement du cerveau biologique. Comme l'ont proposé LeCun et al. dans leur article pionnier de 1998 [13], ces architectures sont composées de multiples couches de neurones interconnectés effectuant des opérations mathématiques pour transformer l'information.

Chaque neurone artificiel reçoit des signaux d'entrée, les pondère, les somme, et produit un signal de sortie en fonction d'une fonction d'activation non-linéaire. Cette structure permet aux RNA d'approximer des fonctions complexes, comme démontré par le théorème d'approximation universelle.

Un type de RNA particulièrement performant pour la vision par ordinateur est le réseau de neurones convolutionnel (CNN), introduit par LeCun et al. [13]. Les CNN comprennent des couches de convolution qui extraient automatiquement des caractéristiques visuelles, suivies de couches de pooling pour réduire la dimensionnalité, et enfin de couches entièrement connectées pour la classification.

Cette architecture innovante a révolutionné de nombreux domaines tels que la classification d'images, comme l'illustre l'étude de Krizhevsky et al. en 2012 [14] qui a établi de nouveaux records de performance sur le défi ImageNet. Grâce à leur capacité d'apprentissage à partir de données, les RNA ont permis des progrès considérables dans des tâches complexes de perception, de compréhension et de prise de décision.

1.5.2 Architecture des réseaux de neurones profonds

Les réseaux de neurones profonds, également appelés DNNs (Deep Neural Networks), sont des architectures particulièrement puissantes et performantes pour résoudre une large gamme de problèmes complexes. Ces modèles sont composés de multiples couches de neurones artificiels qui permettent d'extraire et d'apprendre des représentations hiérarchiques et abstraites des données, les rendant très efficaces pour des tâches d'apprentissage super-

visé ou non supervisé.

Composants d'un réseau de neurones profond Un réseau de neurones profond typique est principalement composé de trois types de couches :

Couche d'entrée Cette couche reçoit les données brutes (images, texte, signaux audio, etc.) qui seront ensuite traitées par le réseau.

Couches cachées Ce sont les couches intermédiaires du réseau, responsables de l'extraction et de l'apprentissage des représentations hiérarchiques des données. Elles sont constituées de neurones interconnectés qui effectuent des opérations mathématiques complexes sur les entrées pour produire des sorties de plus en plus abstraites et significatives.

Couche de sortie Cette dernière couche produit le résultat final du modèle, sous forme de prédiction, de classification, de régression, etc. en fonction de la tâche à résoudre.

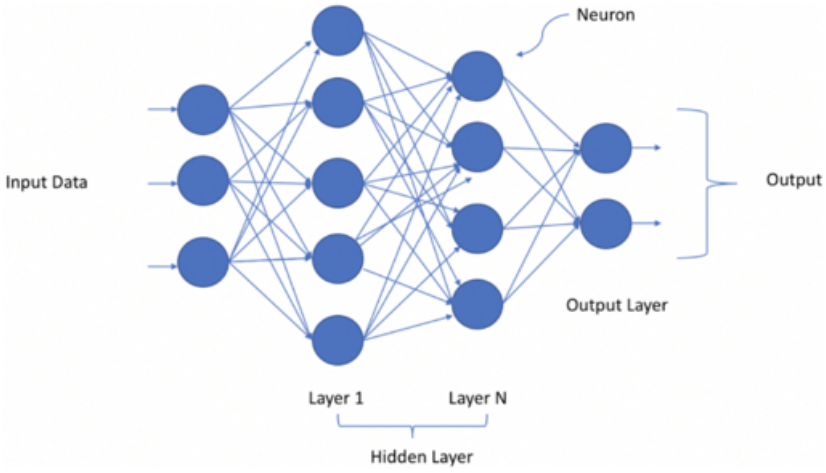


FIGURE 1.1 – Architecture typique d'un réseau de neurones profond [15]

Les réseaux de neurones profonds ont démontré leur capacité à résoudre avec succès une grande variété de problèmes complexes, tels que la reconnaissance d'images, la traduction automatique, la génération de texte, la détection d'anomalies, la robotique, etc. Comme l'expliquent LeCun, Bengio et Hinton (2015), "Les réseaux de neurones profonds ont récemment démontré des performances remarquables dans de nombreuses tâches complexes telles que la reconnaissance d'images, la traduction automatique, la génération de texte, etc. Leur capacité à apprendre des représentations de haut niveau à partir des données a été un facteur clé de leur succès."[16]

Leurs performances surpassent souvent celles des approches traditionnelles de machine learning, notamment grâce à leur aptitude à apprendre des représentations de haut niveau à partir des données. "Les progrès spectaculaires des réseaux de neurones profonds ont permis des avancées dans des domaines variés comme la robotique, la détection d'anomalies, etc. Leur aptitude à extraire automatiquement des features pertinentes à partir des données brutes est un atout majeur par rapport aux approches traditionnelles de machine learning."[17]

Cependant, l'architecture et l'entraînement des DNNs peuvent être très coûteux en res-

sources de calcul et en données, nécessitant parfois des infrastructures matérielles et logicielles spécialisées. C'est un domaine de recherche actif visant à optimiser ces aspects pour rendre les réseaux de neurones profonds plus efficaces et accessibles.

1.5.3 Principaux types de réseaux de neurones profonds (CNN, RNN, LSTM)

Les réseaux de neurones profonds incluent diverses architectures spécialisées pour des tâches spécifiques. Examinons chacune d'entre elles plus en détail :

1.5.3.1 Réseaux de neurones convolutionnels (CNN)

Les réseaux de neurones convolutionnels (CNN) sont une architecture spécifique de réseaux de neurones profonds, souvent utilisée dans le domaine de la vision par ordinateur et de la reconnaissance d'images. Leur conception est inspirée par le fonctionnement du cortex visuel humain.

Voici comment fonctionne un CNN :

1. **Convolution** : La couche de convolution est la pierre angulaire des CNN. Elle consiste en plusieurs filtres (ou noyaux) qui effectuent des opérations de convolution sur l'image d'entrée. Chaque filtre détecte des motifs locaux dans l'image, comme des bords, des textures, ou d'autres caractéristiques importantes.
2. **Pooling (sous-échantillonnage)** : Après la couche de convolution, les CNN utilisent souvent des couches de pooling pour réduire la dimension de la représentation spatiale, tout en préservant les caractéristiques essentielles. Le pooling aide également à rendre la représentation plus invariante aux translations et aux petites déformations dans l'image.
3. **Activation** : Après chaque couche de convolution et de pooling, une fonction d'activation est appliquée, généralement la fonction ReLU (Rectified Linear Unit), qui introduit une non-linéarité dans le réseau.
4. **Réseau de neurones entièrement connecté** : Après plusieurs couches de convolution, de pooling et d'activation, les caractéristiques extraites sont acheminées vers un réseau de neurones entièrement connecté pour la classification finale. Cette partie du réseau agit comme un classifieur pour attribuer des étiquettes aux images en fonction des caractéristiques extraites.

En résumé, les CNN exploitent la structure spatiale des données d'entrée (comme les images) en utilisant des filtres de convolution pour extraire des caractéristiques significatives, puis utilisent des couches de pooling pour réduire la dimensionnalité de la représentation. Cette architecture a démontré une grande efficacité dans la reconnaissance d'objets, la classification d'images et de nombreuses autres tâches de vision par ordinateur.

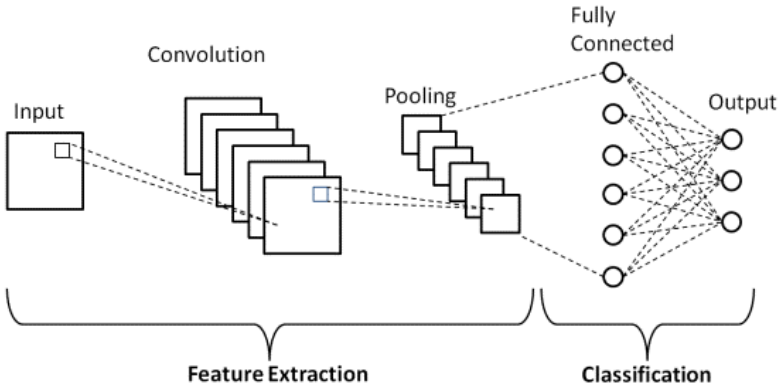


FIGURE 1.2 – Architecture d'un CNN basique [18]

1.5.3.2 Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une architecture spécifique de réseaux de neurones profonds, conçue pour modéliser des séquences de données. Contrairement aux réseaux de neurones classiques, les RNN ont des connexions récurrentes, ce qui leur permet de prendre en compte l'ordre séquentiel des données.

Le fonctionnement détaillé d'un RNN est le suivant :

1. **Connexions récurrentes** : Les RNN ont des connexions récurrentes qui leur permettent de mémoriser des informations à partir des étapes de temps précédentes. À chaque pas de temps t, le réseau prend en compte les données d'entrée x_t ainsi que l'état caché h_{t-1} provenant de l'étape de temps précédente pour calculer un nouvel état caché h_t. Cette capacité à conserver une mémoire à court terme des états précédents est ce qui les rend efficaces pour modéliser des séquences.
2. **Propagation de l'information** : L'information est propagée à travers les étapes de temps du RNN, permettant au réseau de prendre en compte l'ordre séquentiel des données. À chaque pas de temps t, l'état caché h_t est calculé en combinant les données d'entrée x_t avec l'état caché précédent h_{t-1} à l'aide de poids de connexion appris par le réseau.
3. **Longueur de séquence variable** : Contrairement à de nombreux autres types de réseaux neuronaux, les RNN peuvent traiter des séquences de longueurs variables. Cela signifie qu'ils sont capables de gérer des séquences de différentes longueurs, ce qui les rend adaptés à de nombreuses tâches, telles que la traduction automatique, la génération de texte et l'analyse de séquences temporelles.

En résumé, les réseaux de neurones récurrents utilisent des connexions récurrentes pour modéliser des séquences de données, en prenant en compte l'ordre séquentiel des données. Cette architecture est particulièrement utile pour les tâches impliquant des données séquentielles, où la compréhension du contexte temporel est essentielle.

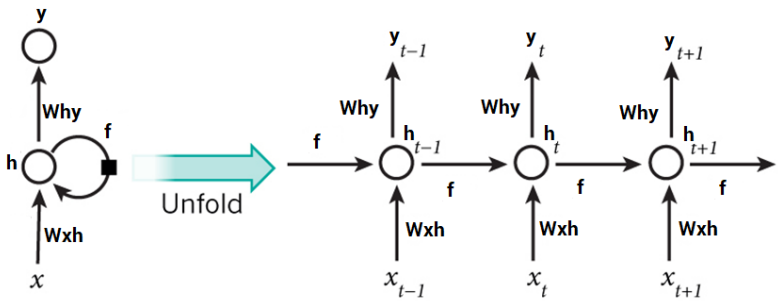


Figure 1.3 – Architecture d'un RNN basique [19]

1.5.3.3 Réseaux LSTM (Long Short-Term Memory)

Les réseaux LSTM (Long Short-Term Memory) sont une variante des réseaux de neurones récurrents (RNN) conçue pour résoudre le problème de disparition du gradient et capturer les dépendances à long terme dans les données séquentielles. Les LSTM ont été largement utilisés dans diverses tâches de traitement du langage naturel, d'analyse de séries temporelles, de reconnaissance vocale, et bien plus encore. Leur architecture comprend des mécanismes spécialisés qui leur permettent de stocker et de récupérer des informations sur de longues séquences.

1.5.3.4 Architecture des LSTM

L'architecture des LSTM repose sur les composants clés suivants :

1. **État de la cellule (C)** : Représente la mémoire des LSTM et peut stocker des informations sur de longues séquences. Il peut être mis à jour, effacé ou lu à chaque pas de temps.
2. **État caché (H)** : L'état caché sert d'intermédiaire entre l'état de la cellule et le monde extérieur. Il peut choisir sélectivement de se souvenir ou d'oublier des informations de l'état de la cellule et produire la sortie.
3. **Porte d'entrée (i)** : Contrôle le flux d'informations dans l'état de la cellule. Il peut apprendre à accepter ou à rejeter les données entrantes.
4. **Porte d'oubli (f)** : Détermine quelles informations de l'état de la cellule précédente doivent être conservées et lesquelles doivent être rejetées. Elle permet aux LSTM d'« oublier » les informations non pertinentes.
5. **Porte de sortie (o)** : Contrôle les informations utilisées pour produire la sortie à chaque pas de temps. Elle décide quelle partie de l'état de la cellule doit être révélée au monde extérieur.

1.5.3.4.1 Types de LSTM

Les LSTM se déclinent en plusieurs variantes :

— **LSTM Vanilla** : Résout le problème de la disparition du gradient, ce qui lui permet

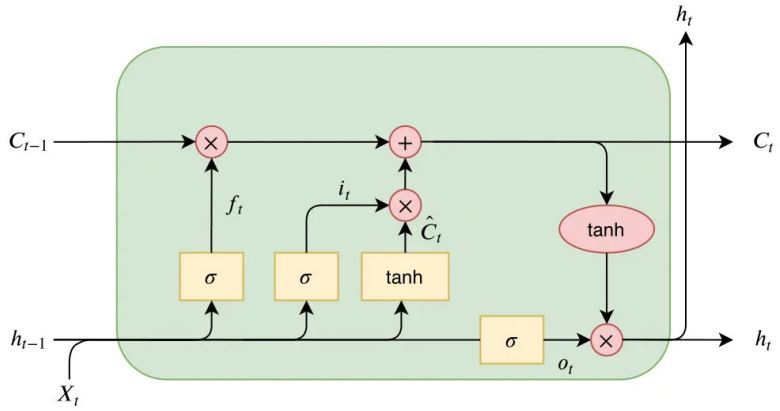


FIGURE 1.4 – Architecture d'un LSTM basique [20]

de capturer des dépendances à longue distance dans les séquences. Cependant, il est plus complexe et nécessite plus de calculs que les RNN traditionnels.

- **LSTM Empilé** : Améliore les performances pour les tâches nécessitant la modélisation de dépendances complexes dans le temps, mais cela augmente la complexité du modèle et le temps d'entraînement.
- **LSTM Bidirectionnel (BiLSTM)** : Capture le contexte à la fois des pas de temps passés et futurs, ce qui le rend efficace pour les tâches où le contexte bidirectionnel est essentiel. Cependant, il nécessite le double de calculs par rapport aux LSTM unidirectionnels.
- **Unité récurrente à portes (GRU)** : Offre une alternative aux LSTM, plus efficace en termes de calcul, tout en résolvant le problème de la disparition du gradient. Cependant, il peut ne pas performer aussi bien que les LSTM dans les tâches avec des dépendances complexes.

les LSTM ont été essentiels dans diverses applications, en particulier dans le traitement du langage naturel, où ils ont été utilisés dans des tâches telles que la traduction automatique, l'analyse des sentiments et la génération de texte. Ils ont également été utilisés dans la prévision de séries temporelles, la reconnaissance vocale et de nombreux autres domaines. Les chercheurs et les ingénieurs continuent d'explorer des variantes des modèles LSTM et des combinaisons avec d'autres architectures.

1.6 Apprentissage par Transfert (Transfer Learning)

L'apprentissage par transfert est une méthode en apprentissage automatique où un modèle développé pour une tâche initiale est réutilisé comme point de départ pour une tâche secondaire différente mais liée. Cette technique est particulièrement utile lorsque les ressources pour entraîner un modèle de zéro sont limitées, ou lorsque les données disponibles pour la nouvelle tâche sont insuffisantes pour entraîner un modèle performant. Comme le soulignent Pan et Yang (2010), "L'apprentissage par transfert permet d'améliorer l'apprentissage dans la tâche cible en transférant les connaissances de la tâche source apparen-

tée"[21]. Cette approche offre de nombreux avantages, notamment une réduction du temps d'entraînement et une amélioration des performances sur des tâches où les données sont rares.

L'apprentissage par transfert trouve ses racines dans la psychologie cognitive, où il est reconnu que les humains appliquent souvent des connaissances acquises dans un domaine à de nouveaux problèmes similaires. Dans le contexte de l'apprentissage automatique, cette technique permet d'exploiter les caractéristiques de bas niveau apprises sur de grands ensembles de données pour améliorer les performances sur des tâches spécifiques avec moins de données. Par exemple, un modèle entraîné sur un grand ensemble d'images naturelles peut être adapté pour reconnaître des types spécifiques de cellules dans des images médicales, même avec un ensemble de données limité.

1.6.1 Concept et Importance

L'apprentissage par transfert constitue un paradigme fondamental en apprentissage automatique, s'appuyant sur le principe que les connaissances acquises lors de la résolution d'une tâche peuvent être exploitées pour améliorer les performances sur une tâche connexe. Ce concept s'inspire de la capacité humaine à généraliser les apprentissages d'un domaine à un autre. Formellement, comme le définissent Torrey et Shavlik, "l'apprentissage par transfert implique l'amélioration de l'apprentissage dans une nouvelle tâche en transférant les connaissances d'une tâche apparentée qui a déjà été apprise" [22].

L'apprentissage par transfert repose sur l'hypothèse que les représentations internes apprises par un modèle sur une tâche source peuvent être utiles pour une tâche cible. Cette hypothèse est soutenue par la théorie de l'apprentissage statistique, qui suggère que les représentations qui capturent les structures sous-jacentes des données sont souvent transférables entre des tâches similaires. Pan et Yang ont formalisé cette idée en définissant un domaine D comme composé d'un espace de caractéristiques X et d'une distribution de probabilité marginale $P(X)$, où $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$ une tâche T est définie par un espace de labels Y et une fonction de prédiction $f(\cdot)$ qui peut être apprise à partir des paires de données d'entraînement $\{x_i, y_i\}$, où $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$.

Dans le contexte des réseaux de neurones profonds, l'apprentissage par transfert se manifeste souvent par la réutilisation des couches inférieures d'un modèle pré-entraîné, qui ont appris à extraire des caractéristiques générales. Par exemple, dans le domaine de la vision par ordinateur, un réseau convolutif entraîné sur un vaste ensemble de données comme ImageNet (qui contient plus de 14 millions d'images annotées) peut être adapté pour une tâche de reconnaissance d'images spécifique avec un ensemble de données limité. Le processus de transfert peut être décomposé en plusieurs étapes :

- **Pré-entraînement :** Le modèle est d'abord entraîné sur une tâche source avec un grand ensemble de données.
- **Transfert :** Les poids des couches inférieures sont conservés et transférés vers le nouveau modèle.
- **Adaptation :** Les couches supérieures sont soit affinées (fine-tuned) avec les données

de la tâche cible, soit remplacées par de nouvelles couches spécifiques à la tâche.

— **Optimisation :** Le modèle est entraîné sur la tâche cible, souvent avec un taux d'apprentissage plus faible pour les couches transférées.

L'importance de l'apprentissage par transfert se manifeste à plusieurs niveaux :

— Efficacité computationnelle : En réutilisant des modèles pré-entraînés, on réduit significativement le temps et les ressources nécessaires pour l'entraînement. Yosinski et al. ont démontré que le transfert peut accélérer la convergence de 2 à 10 fois par rapport à un entraînement à partir de zéro [23].

— Performance sur des ensembles de données limités : Il permet d'obtenir des performances élevées même lorsque les données spécifiques à la tâche sont rares, en exploitant les connaissances générales acquises sur de grands ensembles de données. Kornblith et al. ont montré que les modèles pré-entraînés sur ImageNet surpassent systématiquement les modèles entraînés à partir de zéro sur une variété de tâches de vision par ordinateur, même lorsque les ensembles de données cibles sont relativement grands[24].

— Généralisation améliorée : Les modèles utilisant l'apprentissage par transfert ont souvent une meilleure capacité de généralisation, car ils s'appuient sur des représentations robustes apprises sur des données diversifiées.

— Adaptabilité : Cette approche facilite l'adaptation rapide des modèles à de nouvelles tâches ou domaines, accélérant ainsi le déploiement de solutions d'IA dans divers secteurs.

— Réduction du biais de domaine : L'apprentissage par transfert peut aider à réduire le biais de domaine en permettant aux modèles d'apprendre des représentations plus générales. Cependant, il est important de noter que le transfert peut également propager des biais présents dans les données source, nécessitant une attention particulière lors de l'application à des domaines sensibles.

L'efficacité de l'apprentissage par transfert dans divers domaines, notamment la vision par ordinateur, le traitement du langage naturel et l'apprentissage par renforcement à etais valide par la caummunaute scientifique. Par exemple les caractéristiques apprises par les réseaux convolutifs sur ImageNet sont transférables à une large gamme de tâches de vision par ordinateur, même lorsque les domaines sont significativement différents

Dans le domaine du traitement du langage naturel, l'émergence de modèles de langage pré-entraînés à grande échelle comme BERT a révolutionné l'approche de nombreuses tâches de NLP. Ces modèles, entraînés sur d'énormes corpus de texte, peuvent être facilement adaptés à des tâches spécifiques comme la classification de sentiment, la réponse aux questions, ou la traduction automatique, souvent avec des performances état de l'art.

L'apprentissage par transfert représente une avancée majeure dans le domaine de l'apprentissage automatique, offrant une solution élégante aux problèmes de rareté des données et d'efficacité computationnelle. Son importance continuera probablement à croître à mesure que les modèles deviennent plus complexes et que les applications de l'IA se diversifient.

1.6.2 Types d'Apprentissage par Transfert

1.6.2.1 Feature Extraction (Extraction de Caractéristiques)

Dans cette approche, un modèle pré-entraîné, tel qu'un réseau de neurones convolutif (CNN) comme VGG-16 ou ResNet-50, est utilisé pour extraire des caractéristiques ou des représentations des données. Les couches convolutives du modèle pré-entraîné, qui capturent des caractéristiques de bas et de haut niveau, sont conservées et utilisées pour transformer les nouvelles données en vecteurs de caractéristiques. Ces vecteurs sont ensuite utilisés comme entrée pour un nouveau modèle simple (comme une régression logistique ou un réseau de neurones entièrement connecté peu profond) entraîné pour la nouvelle tâche. L'extraction de caractéristiques permet de tirer parti des connaissances encapsulées dans les couches profondes du modèle pré-entraîné, réduisant ainsi la nécessité de données étiquetées massives et de longues périodes d'entraînement. Par exemple, en utilisant un modèle pré-entraîné sur ImageNet, les caractéristiques extraites peuvent être utilisées pour classifier de nouvelles images avec une précision raisonnable, même lorsque les données d'entraînement sont limitées. Cette approche est particulièrement avantageuse pour des tâches où l'acquisition de données annotées est coûteuse ou difficile.

1.6.2.2 Fine-Tuning

Cette méthode implique de prendre un modèle pré-entraîné et de le réentraîner (fine-tuning) sur la nouvelle tâche. Typiquement, les couches inférieures du modèle, qui capturent des caractéristiques générales, sont maintenues fixes, tandis que les couches supérieures, plus spécifiques à la tâche, sont réentraînées. Par exemple, dans le fine-tuning d'un modèle ResNet, les premières couches convolutives peuvent rester gelées, tandis que les couches de classification finales sont adaptées à la nouvelle tâche. Cette technique permet au modèle de conserver les connaissances acquises sur la tâche initiale tout en s'adaptant à la nouvelle tâche. L'article [25] "Convolutional Neural Networks for Visual Recognition" par Krizhevsky et al. (2012) démontre que le fine-tuning de modèles pré-entraînés sur ImageNet a permis d'obtenir de meilleures performances sur des tâches de classification d'images spécifiques avec peu de données. Le fine-tuning est particulièrement efficace lorsque la nouvelle tâche est similaire à la tâche initiale, car les caractéristiques de bas niveau (telles que les bords et les textures) restent souvent pertinentes. De plus, en ajustant les couches supérieures, le modèle peut apprendre des caractéristiques spécifiques à la nouvelle tâche, améliorant ainsi sa précision et sa robustesse.

1.6.2.3 Domain Adaptation (Adaptation de Domaine)

L'adaptation de domaine vise à adapter un modèle entraîné dans un domaine source à un domaine cible différent mais lié. Cette technique est cruciale lorsque les distributions des données source et cible diffèrent significativement. Les méthodes d'adaptation de domaine comprennent l'apprentissage de représentations invariantes au domaine, la réentraînement sur un mélange de données source et cible, et l'utilisation de techniques adversariales pour

minimiser la divergence entre les distributions des domaines source et cible. Par exemple, des méthodes telles que les Domain-Adversarial Neural Networks (DANNs) utilisent un classificateur adversarial pour encourager l'apprentissage de représentations qui sont indistinguables entre les domaines source et cible. Les réseaux adversariaux génératifs (GANs) peuvent également être utilisés pour générer des exemples dans le domaine cible, aidant ainsi à combler l'écart entre les domaines. Ces techniques ont montré des améliorations notables dans des applications telles que la segmentation d'images médicales où les données annotées peuvent être rares et coûteuses à obtenir. De plus, des méthodes comme la "coral alignment" et l'"MMD (Maximum Mean Discrepancy)" permettent de réduire la différence statistique entre les distributions source et cible, améliorant ainsi la performance du modèle sur le domaine cible.

1.6.3 Processus de l'Apprentissage par Transfert

1.6.3.1 Étape 1 : Sélection d'un Modèle Pré-entraîné

La première étape consiste à choisir un modèle qui a été préalablement entraîné sur une tâche similaire à celle visée. La sélection du modèle dépend de la nature de la nouvelle tâche. Par exemple, pour des tâches de vision par ordinateur, des modèles tels que VGG, ResNet, ou EfficientNet sont couramment utilisés. Pour des tâches de traitement du langage naturel, des modèles comme BERT, GPT, ou RoBERTa sont souvent préférés. Il est crucial de choisir un modèle dont les caractéristiques apprises sont pertinentes pour la nouvelle tâche afin de maximiser les bénéfices du transfert. Par exemple, un modèle pré-entraîné sur ImageNet pour la classification d'images générales serait approprié pour une tâche de classification d'images spécifiques telles que la reconnaissance de types de plantes.

1.6.3.2 Étape 2 : Prétraitement des Données

L'étape suivante implique l'adaptation des données de la nouvelle tâche pour qu'elles soient compatibles avec le modèle pré-entraîné. Pour les images, cela peut inclure des étapes comme la redimension des images à une taille spécifique (par exemple, 224x224 pour ResNet), la normalisation des pixels selon les statistiques du dataset original, et l'application d'augmentations de données pour augmenter la variabilité des données d'entraînement. Pour les textes, le prétraitement peut inclure la tokenisation, la conversion en indices de vocabulaire, et la gestion des séquences de longueur variable en utilisant du padding ou du truncating. Par exemple, pour utiliser BERT, les textes doivent être tokenisés avec le tokenizer de BERT, convertis en tokens d'input IDs, de segment IDs, et de masks d'attention, et les séquences doivent être padées à une longueur fixe.

1.6.3.3 Étape 3 : Ajustement et Réentraînement

— **Feature Extraction :** Cette méthode consiste à utiliser les couches convolutionnelles ou transformeur d'un modèle pré-entraîné pour extraire des caractéristiques des nou-

velles données. Ces caractéristiques extraites sont ensuite utilisées comme entrée pour un nouveau classificateur, tel qu'une couche dense ou une régression logistique, qui est entraîné spécifiquement pour la nouvelle tâche. Par exemple, dans le cadre de la classification d'images, les sorties de la dernière couche convolutive d'un ResNet peuvent être utilisées comme caractéristiques pour entraîner un classificateur SVM.

— **Fine-Tuning :** Cette approche implique de décongeler certaines couches du modèle pré-entraîné et de les réentraîner sur les nouvelles données. Cela permet de mettre à jour les poids des couches pour qu'ils soient adaptés à la nouvelle tâche tout en conservant les caractéristiques générales apprises lors de l'entraînement initial. Typiquement, les premières couches, qui capturent des caractéristiques de bas niveau comme les bords et les textures, sont maintenues fixes, tandis que les couches supérieures, plus spécifiques à la tâche, sont réentraînées. L'article [26] "Efficient Fine-Tuning of Pretrained Models for Text Classification" de Xu et al. (2019) présente des stratégies de fine-tuning efficaces, telles que l'utilisation de taux d'apprentissage différenciés pour différentes couches, l'entraînement progressif des couches, et l'utilisation de régularisation pour éviter le surajustement lors de l'adaptation des modèles pré-entraînés à des tâches de classification de texte spécifiques.

1.6.3.4 Étape 4 : Évaluation et Ajustement

Après le réentraînement, il est essentiel d'évaluer les performances du modèle fine-tuné sur un ensemble de validation ou de test indépendant. Les métriques d'évaluation couramment utilisées comprennent l'exactitude, la précision, le rappel, et le F1-score pour les tâches de classification. Pour les tâches de régression, des métriques comme l'erreur quadratique moyenne (MSE) ou l'erreur absolue moyenne (MAE) sont utilisées. Basé sur ces évaluations, il peut être nécessaire d'ajuster les hyperparamètres du modèle, tels que le taux d'apprentissage, la taille du batch, ou les coefficients de régularisation. De plus, des techniques telles que la validation croisée peuvent être employées pour assurer la robustesse et la généralisation du modèle. L'ajustement des hyperparamètres peut être réalisé à l'aide de méthodes de recherche hyperparamétrique comme la recherche en grille, la recherche aléatoire, ou l'optimisation bayésienne.

1.7 Éthique et biais en Machine Learning

Ces algorithmes d'apprentissage automatique sont de plus en plus utilisés pour automatiser des décisions ayant un impact important sur la vie des individus. Cependant, cette utilisation croissante soulève de sérieuses questions éthiques qu'il est crucial de prendre en compte.

En effet, les algorithmes de ML, du fait de leur complexité et de leur opacité, peuvent engendrer des biais, des discriminations et des erreurs aux conséquences parfois graves. Par exemple, en 2019, le système d'aide à la décision utilisé dans le système de santé américain a été accusé de discriminer les patients noirs, leur attribuant un niveau de risque médical inférieur à celui des patients blancs alors qu'ils étaient en réalité plus malades. Cet exemple

illustre les dérives potentielles lorsque des décisions cruciales sont déléguées à des systèmes algorithmiques opaques.

De plus, l'automatisation de décisions sensibles pose des problèmes de responsabilité en cas de dommages. Enfin, l'utilisation massive de données personnelles soulève des enjeux fondamentaux en termes de confidentialité et de respect de la vie privée.

Il est donc essentiel d'adopter une approche éthique et responsable dans le développement et le déploiement de ces technologies émergentes, afin de maximiser leurs bénéfices tout en minimisant les risques et les dérives potentielles. C'est l'objet de cette section qui explore en détail les principales problématiques éthiques liées à l'utilisation des algorithmes de Machine Learning.

1.7.1 Biais et équité dans les algorithmes de Machine Learning

1.7.1.1 Explication des différents types de biais

Les algorithmes de Machine Learning peuvent être sujets à différents types de biais :

Biais de sélection : lorsque les données d'entraînement ne sont pas représentatives de la population réelle, par exemple si certains groupes sont sous-représentés dans les données. Un exemple typique est l'utilisation de données de reconnaissance faciale qui surreprésentent les visages de personnes blanches, conduisant à de moins bonnes performances pour d'autres groupes ethniques.

Biais de mesure : lorsque les variables utilisées pour entraîner le modèle ne mesurent pas correctement le phénomène étudié. Par exemple, utiliser le niveau d'éducation comme proxy pour estimer la capacité d'un candidat à l'emploi pourrait désavantager les personnes issues de milieux défavorisés.

Biais algorithmiques : lorsque les choix de conception et d'implémentation des algorithmes introduisent des biais, par exemple en favorisant certains groupes. Suresh et Guttag (2019) donnent l'exemple d'un algorithme d'attribution de crédits qui accorderait des taux plus avantageux aux hommes qu'aux femmes, même sans que cela ne soit intentionnel.

Biais humains : lorsque les concepteurs et développeurs injectent involontairement leurs propres biais dans les algorithmes, en raison de leurs propres préjugés inconscients.

1.7.1.2 Impacts potentiellement discriminatoires des biais

Ces différents types de biais peuvent avoir des conséquences graves dans des domaines sensibles comme la justice, l'emploi ou la santé.

Dans le domaine judiciaire, Angwin et al. (2016) ont montré qu'un système d'aide à la décision utilisé aux États-Unis pour prédire le risque de récidive était biaisé contre les personnes de couleur, leur attribuant des scores de risque plus élevés que les individus blancs à infractions égales. Cela pouvait conduire à des décisions de justice plus sévères

envers certaines communautés.

En matière d'emploi, Lambrecht et Tucker (2019) ont mis en évidence que des annonces pour des postes dans les domaines STEM étaient plus souvent montrées à des hommes qu'à des femmes, même quand le ciblage publicitaire était censé être neutre. Ce biais de genre dans l'affichage des annonces pouvait désavantager les candidates féminines.

Dans le domaine de la santé, Chen et al. (2019) ont montré qu'un algorithme prédictif utilisé pour allouer des soins de santé aux patients chroniques attribuait un niveau de risque médical inférieur aux patients noirs par rapport aux patients blancs, alors qu'ils étaient en réalité plus malades. Cela pouvait conduire à des prises en charge différenciées et discriminatoires.

Ces exemples illustrent bien les impacts potentiellement graves et discriminatoires que peuvent avoir les biais algorithmiques dans des domaines à fort enjeu éthique.

1.7.1.3 Techniques pour détecter et atténuer les biais algorithmiques

Plusieurs techniques existent pour identifier et réduire les biais dans les algorithmes de Machine Learning :

- Audits algorithmiques : analyse approfondie des algorithmes pour détecter la présence et l'origine des biais.
- Rééquilibrage des données d'entraînement : techniques de ré-échantillonnage pour corriger les déséquilibres dans les données.
- Ajustement des hyperparamètres : optimisation des paramètres des algorithmes pour minimiser les biais.
- Explainability et interprétabilité : techniques d'IA explicable permettant de comprendre les décisions des modèles.
- Tests d'équité : évaluation des impacts différenciés des algorithmes sur différents groupes.

L'application de ces techniques, en complément d'une réflexion éthique approfondie, est essentielle pour développer des systèmes de Machine Learning équitables et non discriminatoires.

1.7.2 Biais algorithmiques et implications sociétales

Les biais algorithmiques ont des implications importantes dans divers domaines de la société, notamment :

Accès équitable aux opportunités : Les décisions automatisées basées sur des modèles biaisés peuvent restreindre l'accès aux opportunités telles que l'emploi, le logement, l'éducation et les services financiers. Par exemple, si un algorithme de prêt favorise systématiquement les candidats issus de certaines communautés au détriment d'autres, il perpétue les inégalités économiques et sociales.

Renforcement des stéréotypes : Les modèles de Machine Learning peuvent renforcer les stéréotypes existants en favorisant certains groupes par rapport à d'autres. Par exemple, dans les systèmes de recommandation de produits, un algorithme peut recommander des jouets pour garçons à des garçons et des jouets pour filles à des filles, renforçant ainsi les normes de genre traditionnelles.

Confiance et légitimité des institutions : Lorsque les décisions importantes sont prises de manière opaque et injuste par des algorithmes, cela peut saper la confiance du public dans les institutions qui les utilisent. Par exemple, si un système de justice prédictive est perçu comme discriminatoire envers certaines communautés, cela pourrait remettre en question l'intégrité du système judiciaire dans son ensemble.

1.7.3 Mesures pour atténuer les biais et garantir l'équité dans les modèles

Pour atténuer les biais algorithmiques et leurs implications sociétales, plusieurs approches peuvent être adoptées :

Diversification des données d'entraînement : Il est essentiel d'utiliser des ensembles de données diversifiés et représentatifs de la population concernée afin de réduire les biais de représentation.

Transparence et responsabilité : Les organisations doivent rendre compte de l'utilisation des algorithmes et de leurs impacts sur la société, en mettant en œuvre des mécanismes de transparence et de responsabilité.

Évaluation continue : Les modèles de Machine Learning doivent être continuellement évalués pour détecter et corriger les biais potentiels. Cela nécessite des mécanismes robustes de surveillance et de rétroaction.

Conception éthique : Les concepteurs d'algorithmes doivent intégrer des principes éthiques dès la phase de conception, en veillant à ce que les modèles soient équitables, transparents et responsables.

En adoptant ces stratégies, il est possible de développer et de déployer des systèmes de Machine Learning plus équitables et plus inclusifs, contribuant ainsi à une société plus juste et plus harmonieuse.

1.8 Conclusion

La compréhension du machine et du deep learning ainsi que les réseaux de neurones profonds et de leurs applications. Et la sensibilisation aux enjeux éthiques et aux biais algorithmiques, sont essentielles pour développer des systèmes de Machine Learning équitables et performant. Ce chapitre a etait une introduction aux concepts de base du machine learning et du deep learning les différentes architectures spécialisées comme les CNN, les RNN

et les LSTM. De plus, il a examiné les questions éthiques importantes liées à l'utilisation de l'IA, notamment les biais algorithmiques et leurs implications sociétales.

Il est crucial pour les praticiens de l'IA de prendre conscience de ces enjeux et d'adopter des pratiques de conception éthiques tout au long du processus de développement des modèles. En intégrant la diversité des données, la transparence et la responsabilité dans la conception des algorithmes, il est possible de réduire les risques de discrimination et de favoriser une utilisation équitable et inclusive de l'IA dans la société.

Dans le prochain chapitre, nous plongerons dans le contexte qui a conduit à l'idée de créer un système de screening de CV. Ensuite, nous aborderons l'organisme d'accueil où le projet a pris forme. Enfin, nous plongerons dans le processus de développement rigoureux qui a été adopté pour concrétiser cette vision.

.

Chapitre 2

Contexte du projet et méthodologie de conception

2.1 Introduction

Dans ce chapitre, nous nous attacherons à présenter de manière exhaustive le contexte de notre projet de recherche et développement, qui porte sur l'utilisation de l'apprentissage profond pour automatiser l'extraction d'informations à partir de curriculum vitae (CV). Cette problématique s'inscrit dans un cadre plus large de transformation numérique des processus de recrutement et de gestion des ressources humaines.

Nous commencerons par exposer en détail le contexte général du projet, en mettant en lumière les enjeux actuels liés à l'analyse des CV et les opportunités offertes par les technologies d'intelligence artificielle. Nous aborderons notamment les défis rencontrés par les entreprises face à l'afflux massif de candidatures et la nécessité de traiter efficacement ces informations.

Ensuite, nous présenterons de manière approfondie l'entreprise Tech Instinct, qui nous accueille pour la réalisation de ce stage de recherche appliquée. Nous détaillerons son positionnement sur le marché, ses domaines d'expertise, ainsi que sa culture d'entreprise axée sur l'innovation et l'agilité. Cela permettra de mieux comprendre le cadre dans lequel s'inscrit notre projet et les synergies potentielles avec les compétences de l'entreprise.

Par la suite, nous aborderons de façon détaillée la problématique centrale de ce projet. Nous analyserons les enjeux spécifiques liés à l'extraction automatisée d'informations à partir des CV, tels que la diversité des formats, la variabilité du contenu, ou encore la nécessité de prendre en compte les spécificités linguistiques et culturelles. Nous expliquerons comment la conception et l'implémentation de méthodes d'automatisation performantes peuvent permettre d'extraire de manière fiable et exhaustive les informations pertinentes contenues dans les CV, comme les expériences professionnelles, les compétences, la formation, etc. Nous mettrons en évidence les bénéfices attendus en termes d'efficacité et de qualité pour les processus de recrutement et de gestion des ressources humaines.

Enfin, nous définirons de manière précise le processus de recherche et de développement entrepris pour ce projet. Nous expliciterons les différentes étapes méthodologiques suivies, en détaillant pour chacune d'elles les objectifs, les outils et les techniques utilisés. Nous aborderons notamment les phases d'analyse des besoins, de collecte et de préparation des données, de conception des modèles d'apprentissage profond, d'implémentation, de tests et d'évaluation des performances.

2.2 Contexte du projet

Ce projet de recherche et développement s'inscrit dans un contexte actuel marqué par une transformation numérique profonde et rapide des processus de recrutement et de gestion des ressources humaines. Cette évolution est motivée par plusieurs facteurs convergents qui rendent nécessaire l'adoption de nouvelles approches technologiques.

Tout d'abord, nous assistons à une explosion du volume de curriculum vitae à analyser. Cette augmentation est due à plusieurs facteurs : la digitalisation des processus de candidature qui facilite l'envoi de CV, la multiplication des plateformes de recrutement en ligne, ou encore l'internationalisation du marché du travail qui élargit le bassin de candidats potentiels. Face à cet afflux massif de candidatures, les entreprises se trouvent confrontées à un défi de taille : comment traiter efficacement et équitablement l'ensemble de ces dossiers ? Parallèlement, la nécessité d'identifier rapidement les profils les plus pertinents s'est accentuée. Dans un marché du travail de plus en plus compétitif, la rapidité de réaction est devenue un avantage concurrentiel majeur pour les entreprises. Celles-ci doivent être capables de repérer et de contacter les meilleurs talents avant leurs concurrents, ce qui implique une analyse rapide et précise des CV reçus.

Ces défis ont conduit de nombreuses entreprises à s'intéresser aux technologies d'apprentissage profond pour automatiser l'extraction d'informations à partir des CV. L'apprentissage profond, une branche de l'intelligence artificielle, offre en effet des perspectives prometteuses pour traiter efficacement de grands volumes de données textuelles non structurées, comme c'est le cas des CV. Le traitement manuel des CV, bien que toujours largement répandu, montre de plus en plus ses limites face à ces nouveaux enjeux. Il s'avère non seulement chronophage, mobilisant des ressources humaines importantes, mais aussi sujet à des erreurs et à des biais cognitifs. La fatigue, la subjectivité ou encore la pression du temps peuvent conduire à une analyse inconsistante ou incomplète des dossiers, potentiellement préjudiciable tant pour les entreprises que pour les candidats.

L'automatisation de cette tâche, grâce à des méthodes d'apprentissage automatique, ouvre la voie à des gains significatifs en termes d'efficacité et de fiabilité dans les processus de recrutement. Elle permet notamment :

— Un traitement rapide d'un grand volume de CV, réduisant considérablement le temps nécessaire à l'analyse des candidatures.

— Une extraction systématique et exhaustive des informations pertinentes, garantissant une analyse homogène de tous les dossiers. Une réduction des biais humains dans la première phase de sélection des candidats.

— Une amélioration de l'expérience candidat grâce à des retours plus rapides et personnalisés.
— Une optimisation des ressources humaines, permettant aux recruteurs de se concentrer sur des tâches à plus forte valeur ajoutée comme l'évaluation approfondie des candidats présélectionnés.

C'est dans ce contexte dynamique et porteur que s'inscrit notre projet de recherche et développement. Il vise à concevoir et à implémenter des solutions innovantes d'extraction automatique d'informations à partir de curriculum vitae, en s'appuyant sur les dernières avancées dans le domaine de l'apprentissage profond.

Notre ambition est de développer des modèles capables de :

— Reconnaître et extraire avec précision les différentes sections d'un CV (expérience professionnelle, formation, compétences, etc.), indépendamment de leur format ou de leur structure.
— Analyser le contenu de ces sections pour en extraire les informations clés (dates, intitulés de poste, noms d'entreprises, diplômes, compétences techniques, etc.).
— S'adapter à différentes langues et formats de CV, reflétant la diversité des candidatures dans un marché du travail mondialisé.

L'objectif final est de faciliter et d'accélérer les processus de recrutement, tout en améliorant la qualité de l'analyse des dossiers. Cela permettra aux entreprises de prendre des décisions de recrutement plus rapides, plus éclairées et potentiellement plus justes, tout en offrant une meilleure expérience aux candidats.

2.3 Organisme d'accueil

Ce projet de recherche et développement est mené en collaboration étroite avec l'entreprise Tech Instinct, un cabinet de conseil et de réalisation IT innovant basé à Béjaïa, en Algérie. Fondée sur les principes de l'agilité et de l'innovation continue, Tech Instinct s'est positionnée comme un acteur clé dans l'accompagnement des TPE, PME et startups dans leur transformation numérique.

Tech Instinct se distingue par sa capacité à offrir un conseil technologique de pointe, adapté aux besoins spécifiques de chaque client. L'entreprise a développé une expertise reconnue dans quatre domaines clés du développement informatique :

— Front-end et mobile : Tech Instinct conçoit et développe des interfaces utilisateur intuitives et performantes, aussi bien pour des applications web que mobiles. Leurs équipes maîtrisent les derniers frameworks et technologies front-end (React, Angular, Vue.js, React Native, Flutter, etc.) pour créer des expériences utilisateur optimales.
— Back-end et API : L'entreprise excelle dans la conception et le développement de systèmes back-end robustes et évolutifs. Ses développeurs sont versés dans différents langages et frameworks (Node.js, Python, Java, .NET, etc.) et ont une expertise particulière dans la création d'API RESTful et GraphQL pour assurer une communication efficace entre les différentes couches applicatives.

— DevOps et cloud : Tech Instinct a développé une forte compétence dans les pratiques DevOps et le déploiement d'applications sur le cloud. L'entreprise maîtrise les outils d'intégration et de déploiement continus (CI/CD), ainsi que les principales plateformes cloud, permettant d'assurer la scalabilité et la fiabilité des solutions développées.
— Intelligence Artificielle et Analyse de Données : Bien que plus récent, ce domaine d'expertise est en pleine expansion chez Tech Instinct. L'entreprise investit dans le développement de compétences en machine learning et en traitement du langage naturel, ce qui s'aligne parfaitement avec les objectifs de notre projet de recherche.

La culture d'entreprise de Tech Instinct est profondément ancrée dans l'innovation continue et la veille technologique. L'entreprise encourage ses consultants à se former constamment aux dernières technologies et méthodologies, créant un environnement propice à l'expérimentation et à l'adoption de solutions innovantes.

Les principes agiles sont au cœur de l'approche de Tech Instinct, tant dans sa gestion interne que dans sa collaboration avec les clients. L'entreprise pratique des méthodologies comme Scrum, favorisant une communication transparente, une adaptation rapide aux changements et une livraison continue de valeur.

Cette collaboration avec Tech Instinct est un élément clé de la réussite de notre projet, combinant expertise technique, culture de l'innovation et compréhension des enjeux business liés au recrutement et à la gestion des talents.

2.4 Objectif du projet et la problématique

L'analyse manuelle d'un grand volume de CV s'avère aujourd'hui de plus en plus chronophage et sujette à des erreurs. Selon une étude menée par le cabinet de conseil McKinsey, les entreprises consacrent en moyenne 23% de leur temps de recrutement à la phase d'évaluation des candidatures [27]. Une enquête de Capterra révèle que 75% des recruteurs estiment passer trop de temps sur cette tâche[28].

L'identification rapide des profils les plus qualifiés est pourtant un enjeu crucial, dans un contexte de pénurie de talents et de compétition accrue pour attirer les meilleurs candidats. Selon une étude menée par le Boston Consulting Group, 45% des entreprises dans le monde rencontrent des difficultés à pourvoir certains postes[29]. Dans le même temps, le volume de CV à analyser ne cesse d'augmenter, avec la multiplication des candidatures en ligne. Et désormais les réseaux sociaux et les sites d'emploi sont les lieux incontournable pour trouver des talents.

L'automatisation de cette tâche, grâce à des techniques d'apprentissage profond, permettrait ainsi de gagner en efficacité et en fiabilité. Selon une étude menée par le Laboratoire de Recherche en Informatique de l'Université Paris-Saclay, l'utilisation de méthodes d'intelligence artificielle pour l'extraction d'informations à partir de CV permettrait de réduire le temps consacré à cette activité de 40% en moyenne [30].

Plus spécifiquement, il s'agira de développer des solutions capables d'extraire de ma-

nière précise et exhaustive les différentes informations clés contenues dans les CV, telles que les expériences professionnelles, les compétences techniques et comportementales, la formation, etc. Cela permettra non seulement d'accélérer les processus de recrutement, mais également d'améliorer la qualité de l'analyse des dossiers.

La problématique sera donc d'explorer les dernières avancées dans le domaine de l'apprentissage profond appliqué à l'extraction d'informations à partir de documents textuels non structurés, afin de concevoir et d'implémenter des méthodes innovantes répondant aux enjeux spécifiques du traitement des CV.

2.5 Processus du développement

Afin de répondre à la problématique centrale de ce projet de recherche et développement, un processus itératif en plusieurs étapes a été mis en place.

2.5.1 Étape 1 : Étude de l'état de l'art

La première étape a consisté en une recherche approfondie sur les différents algorithmes et méthodes d'extraction d'informations à partir de documents textuels non structurés. Cette phase cruciale avait pour objectif de nous familiariser avec les dernières avancées dans le domaine de l'apprentissage profond (Deep Learning) et du traitement du langage naturel (NLP), afin d'identifier les approches les plus prometteuses pour le traitement spécifique des CV.

Nous avons commencé par des revues exhaustive de la littérature scientifique, en consultant des articles de journaux, des conférences spécialisées, et des thèses de doctorat. Des bases de données comme Google Scholar, IEEE Xplore, et PubMed ont été largement utilisées pour accéder aux publications récentes et pertinentes. Parmi les méthodes explorées, les architectures de réseaux de neurones convolutifs (CNN), et les modèles transformateurs (Transformers) ont été particulièrement étudiées. Et leurs variantes ont été analysés pour leur capacité à comprendre et traiter les textes non structurés.

Les critères de sélection des méthodes comprenaient la précision de l'extraction, la robustesse des modèles face à des variations de format et de langue dans les CV, ainsi que l'efficacité computationnelle. Nous avons également pris en compte des facteurs comme la facilité d'intégration dans des systèmes existants et la scalabilité des solutions proposées.

Lors des sessions de suivi quotidiennes (daily), l'équipe a régulièrement discuté des progrès réalisés dans cette phase de recherche. Ces réunions ont permis d'identifier les pistes les plus prometteuses et de débattre des premières hypothèses sur les méthodes à concevoir. Par exemple, nous avons discuté de l'efficacité des modèles basés sur l'attention, comme BERT, par rapport aux approches plus traditionnelles basées sur des règles ou des algorithmes de machine learning moins sophistiqués.

Ces discussions ont également porté sur les limitations des approches existantes, telles

que les défis posés par les variations de format dans les CV, les ambiguïtés linguistiques, et les erreurs possibles dues à la qualité des données. Les échanges ont conduit à la formulation de questions de recherche spécifiques et à la définition de critères de performance pour évaluer les futures implémentations.

En résumé, cette phase d'étude de l'état de l'art a permis de dresser un panorama complet des technologies disponibles et de poser les bases théoriques nécessaires à la conception des méthodes d'extraction d'informations adaptées aux CV. Elle a également servi de point de départ pour des expérimentations pratiques et des itérations successives, visant à affiner et à valider les solutions proposées.

2.5.2 Étape 2 : Constitution d'un jeu de données

Sur la base des connaissances acquises lors de l'étape précédente, une phase cruciale de collecte et de préparation d'un jeu de données de CV a été entreprise. L'objectif était de constituer un corpus représentatif et diversifié, annoté manuellement, qui servirait à l'entraînement et à l'évaluation des futurs modèles d'extraction d'informations.

La première étape de cette phase a consisté à identifier et à rassembler des CV provenant de diverses sources. Nous avons collecté des CV de bases de données internes d'entreprises partenaires, et de contributions volontaires de candidats. L'objectif était d'obtenir un échantillon varié, reflétant une large gamme de secteurs d'activité, de niveaux d'expérience, de formats de présentation.

L'étape suivante a consisté en l'annotation manuelle des CV. Une équipe d'annotateurs, formée aux spécificités des informations pertinentes à extraire (telles que les expériences professionnelles, les compétences, la formation, etc.), a été mobilisée. Chaque annotateur a utilisé une interface dédiée pour marquer les sections pertinentes de chaque CV, en suivant des directives strictes pour assurer la cohérence et la qualité des annotations.

Pour faciliter et accélérer le processus d'annotation, nous avons utilisé des outils de gestion de données et des plateformes collaboratives. Ces outils ont permis de suivre l'avancement des annotations, de gérer les tâches de manière efficace, et de garantir la qualité grâce à des mécanismes de double annotation et de vérification croisée.

Un autre défi important était d'assurer l'équilibrage des données. Nous avons veillé à inclure des CV de différentes industries, de niveaux d'expérience variés, et de différents formats (chronologique, fonctionnel, mixte) pour que le modèle puisse généraliser efficacement. Des techniques de suréchantillonnage et de sous-échantillonnage ont été utilisées pour ajuster la distribution des classes et éviter les biais.

En résumé, la constitution d'un jeu de données de CV a été une étape complexe et minutieuse, nécessitant une planification rigoureuse et une exécution méthodique. Cette étape a permis de disposer d'un corpus riche et diversifié, essentiel pour l'entraînement et l'évaluation des modèles d'extraction d'informations. Le jeu de données ainsi constitué a posé les bases solides pour les phases suivantes de conception et d'implémentation des méthodes d'extraction

2.5.3 Étape 3 : Conception des méthodes d'extraction

Fort de l'état de l'art réalisé et du jeu de données constitué, l'étape suivante a consisté à concevoir les méthodes d'extraction d'informations les plus adaptées pour notre projet. Cette phase a impliqué un travail approfondi de sélection des architectures de réseaux de neurones, ainsi que la définition des stratégies d'entraînement et d'optimisation des modèles.

Le choix des architectures s'est basé sur plusieurs critères, notamment la capacité des modèles à reconnaitre les sections des CVs et aussi a gérer des sorties qui reussisent a capturer la semantique des CVs, leur robustesse face aux variations de format et de langue, et leur performance en termes de précision et de rappel. Nous avons également pris en compte les ressources computationnelles disponibles et la faisabilité d'intégration des modèles dans des systèmes existants.

Une fois les architectures sélectionnées, nous avons défini des stratégies d'entraînement spécifiques pour optimiser les performances des modèles. Cela a inclus la pré-formation des modèles sur des corpus généraux de texte avant de les affiner sur notre jeu de données annoté de CV. Des techniques d'augmentation de données ont été utilisées pour enrichir le corpus d'entraînement et améliorer la robustesse des modèles. Nous avons également expérimenté avec différentes configurations d'hyperparamètres pour maximiser l'efficacité des modèles.

En résumé, cette étape de conception des méthodes d'extraction a été caractérisée par une approche méthodique et collaborative, combinant des évaluations théoriques et des expérimentations pratiques pour aboutir à des solutions innovantes et efficaces pour l'extraction d'informations à partir de CV.

2.5.4 Étape 4 : Implémentation et tests

Une fois les méthodes d'extraction conçues et optimisées, l'étape suivante a consisté à passer à l'implémentation et à la phase de tests rigoureux. Cette phase a été cruciale pour transformer les concepts théoriques en solutions pratiques et fonctionnelles, prêtes à être intégrées dans l'environnement opérationnel de l'entreprise.

L'implémentation a été réalisée en suivant les bonnes pratiques de développement logiciel, incluant la gestion des dépendances, la modularité du code faciliter la maintenance future.

Parallèlement au développement, une série de tests ont été planifiés et exécutés. Les tests ont évalué la performance globale du système sur des scénarios représentatifs de l'utilisation réelle. Des jeux de données de validation distincts de ceux utilisés pour l'entraînement ont été employés pour garantir la généralisation des modèles.

Les tests ont été complétés par des évaluations de la robustesse du système face à diverses variations et anomalies potentielles dans les CV, telles que des formats différents. Des métriques de performance spécifiques à chaque composant, telles que la vitesse d'ex-

traction et la précision des résultats, ont été mesurées et analysées pour identifier les zones nécessitant des améliorations.

Durant cette phase, les membres de l'équipe ont collaboré étroitement pour résoudre rapidement les problèmes identifiés et optimiser les performances du système. Des sessions de révision régulières ont permis de partager les résultats des tests, de discuter des ajustements nécessaires, et de prioriser les prochaines étapes de développement.

En résumé, l'étape d'implémentation et de tests a représenté une phase critique du projet, transformant les concepts théoriques en solutions concrètes et fonctionnelles grâce à une approche méthodique et collaborative.

Ce processus itératif, alliant recherche, conception, implémentation et tests, a permis de répondre de manière pragmatique et progressive à la problématique initiale du projet.

2.6 Conclusion

Ce chapitre a présenté le processus de développement mis en place pour répondre à la problématique centrale de ce projet, à savoir la conception et l'implémentation de méthodes performantes d'extraction automatique d'informations à partir de curriculum vitae.

Le processus itératif décrit, alliant étude de l'état de l'art, constitution d'un jeu de données, conception des méthodes d'extraction, implémentation et tests, puis déploiement et intégration, a permis de progresser de manière pragmatique et efficace vers la réalisation des objectifs fixés.

Les sessions de suivi quotidiennes (daily) ont joué un rôle essentiel tout au long de ce projet, en offrant un cadre de discussion et de prise de décision éclairée sur les différentes étapes.

Le prochain chapitre sera consacré à l'exploration et à l'implémentation de la première brique de cette solution a savoir les techniques d'apprentissage profond, telles que les architectures Faster R-CNN et Mask R-CNN, pour identifier et segmenter automatiquement les différentes sections d'un CV. Cela permettra d'affiner davantage l'extraction des informations clés contenues dans ces documents non structurés.

Chapitre 3

État de l'Art et Avancées dans le Parsing de CVs : Approches Traditionnelles et Deep Learning

3.1 Introduction

Le parsing de CVs est essentiel pour extraire rapidement des informations clés à partir de documents diversifiés. Ce document explore deux approches principales : les méthodes traditionnelles basées sur des règles et des modèles statistiques, ainsi que les techniques avancées de deep learning comme les réseaux de neurones et les modèles transformers (BERT). L'objectif est de comparer ces méthodes pour comprendre comment elles optimisent le traitement automatisé des CVs. Nous examinerons les défis, les méthodologies, les applications et les performances de chaque approche pour offrir une vue d'ensemble complète des avancées dans ce domaine.

3.2 Revue de Littérature sur le Parsing de CVs

3.2.1 Approches Traditionnelles

Les approches traditionnelles de parsing de CV peuvent être divisées en deux catégories principales :

3.2.1.1 Techniques basées sur des règles

Les techniques de parsing de CV basées sur des règles, décrites par Sarawagi (2008)[31], reposent sur la définition manuelle d'un ensemble de règles et de motifs spécifiques pour identifier et extraire les informations clés des CV. Ces règles sont construites à partir de

l'expertise des spécialistes du domaine et de l'analyse approfondie de la structure et du formatage typiques des CV.

Par exemple, pour extraire le nom d'un candidat, une règle pourrait être : "Si le texte est en gras et situé en haut du document, alors c'est probablement le nom du candidat". Ces règles peuvent ensuite être implémentées sous forme d'expressions régulières, de patrons de reconnaissance d'entités nommées ou d'autres algorithmes similaires.

L'avantage principal de cette approche est qu'elle permet un contrôle fin sur le processus d'extraction, en se basant sur une compréhension approfondie des structures de CV. Cependant, elle nécessite un important travail manuel pour définir les règles, et peut manquer de robustesse face à la diversité des formats de CV.

3.2.1.2 Techniques statistiques

Les techniques statistiques de parsing de CV, décrites par Cortez et al. (2010)[32], s'appuient sur des modèles probabilistes et d'apprentissage automatique pour identifier et extraire les informations pertinentes. Contrairement aux approches basées sur des règles, ces méthodes n'exigent pas de définir manuellement des règles complexes, mais s'appuient plutôt sur l'apprentissage à partir de données annotées.

Par exemple, on peut utiliser des modèles de Markov cachés (HMM) ou des champs aléatoires conditionnels (CRF) pour étiqueter les différents segments d'un CV (nom, expérience, éducation, etc.) en se basant sur les caractéristiques lexicales, structurelles et contextuelles du texte. Ces modèles statistiques peuvent être entraînés sur un corpus de CV annotés manuellement.

L'avantage principal de cette approche est qu'elle peut s'adapter plus facilement à la diversité des formats de CV, en apprenant à partir des données plutôt que de s'appuyer sur des règles prédéfinies. Cependant, elle nécessite un important travail d'annotation des données d'entraînement, et peut manquer de transparence par rapport aux techniques basées sur des règles.

3.2.2 Techniques Modernes d'Apprentissage Automatique

Ces dernières années, les techniques d'apprentissage automatique, et plus particulièrement les modèles de deep learning, ont connu des progrès significatifs dans de nombreuses tâches de traitement du langage naturel, y compris le parsing de CV. Ces approches permettent de surmonter certaines limites des méthodes traditionnelles basées sur des règles ou des modèles statistiques.

Contrairement aux techniques règles, les modèles d'apprentissage automatique n'ont pas besoin d'être programmés manuellement. Ils apprennent à partir de données d'entraînement annotées, ce qui leur permet de s'adapter à la diversité des formats de CV et de capturer des motifs complexes dans les données. De plus, ces modèles ont montré de meilleures performances que les approches traditionnelles sur de nombreuses tâches d'extraction d'in-

formation à partir de textes.

Les principales familles de modèles d'apprentissage automatique utilisées pour le parsing de CV sont les réseaux de neurones convolutifs (CNNs), les réseaux de neurones récurrents (RNNs) et les modèles Transformers comme BERT. Nous allons examiner plus en détail chacune de ces approches.

3.2.2.1 Réseaux de Neurones Convolutifs (CNNs)

Dans le contexte du parsing de CVs, les CNNs peuvent être utilisés pour identifier et extraire les informations clés en traitant le texte comme une séquence de mots ou de caractères. Les CNNs utilisent des filtres de convolution qui balayent la séquence d'entrée pour capturer des motifs locaux. Ces motifs peuvent correspondre à des caractéristiques lexicales ou syntaxiques utiles pour l'extraction d'information.

Par exemple, un filtre de convolution pourrait apprendre à reconnaître des motifs tels que des dates, des noms de lieux, ou des titres de poste. Les sorties des filtres de convolution sont ensuite transmises à une couche de pooling, qui réduit la dimensionnalité des données en ne conservant que les caractéristiques les plus pertinentes. Enfin, une couche de sortie dense utilise ces caractéristiques pour prédire les étiquettes d'extraction d'information.

Les CNNs présentent plusieurs avantages pour le parsing de CVs. Ils peuvent capturer des motifs locaux importants dans le texte, et sont relativement robustes face à la variabilité de la longueur des CVs. Cependant, ils peuvent avoir du mal à capturer des dépendances à long terme dans le texte, ce qui peut limiter leurs performances sur certaines tâches d'extraction d'information.)[33],

3.2.2.2 Réseaux de Neurones Récurrents (RNNs)

Les Réseaux de Neurones Récurrents (RNNs) sont une autre catégorie de modèles de deep learning largement utilisés dans le traitement du langage naturel. Contrairement aux CNNs, les RNNs sont conçus pour traiter des séquences de longueur variable, ce qui les rend particulièrement adaptés aux tâches de parsing de CVs.

Dans le contexte du parsing de CVs, les RNNs peuvent être utilisés pour lire le texte séquentiellement, mot par mot ou caractère par caractère, et mettre à jour leur état interne à chaque étape en fonction de l'entrée précédente. Cette capacité à mémoriser l'information précédente permet aux RNNs de capturer des dépendances à long terme dans le texte, ce qui est crucial pour comprendre la structure et le contexte des CVs.

Par exemple, un RNN pourrait apprendre à reconnaître une séquence de mots correspondant à une expérience professionnelle, même si cette séquence est séparée par plusieurs lignes dans le CV. Les sorties du RNN peuvent ensuite être utilisées pour prédire les étiquettes d'extraction d'information.

Les RNNs présentent plusieurs avantages pour le parsing de CVs. Ils peuvent capturer des dépendances à long terme dans le

texte, et sont capables de traiter des séquences de longueur variable. Cependant, ils peuvent être difficiles à entraîner sur de longues séquences en raison du problème de disparition du gradient, et peuvent manquer de parallélisme comparé aux CNNs.[34]

3.2.2.3 Transformers et BERT

Les Transformers sont une architecture qui a révolutionné le traitement du langage naturel. Contrairement aux RNNs, les Transformers utilisent une attention mécanisme pour traiter l'ensemble de la séquence d'entrée en parallèle, ce qui permet de capturer des dépendances à long terme tout en étant plus efficace à entraîner.

BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langage basé sur les Transformers qui a été pré-entraîné sur un grand corpus de texte. Il a montré des performances exceptionnelles sur de nombreuses tâches de traitement du langage naturel, y compris le parsing de CVs.

Dans le contexte du parsing de CVs, BERT peut être utilisé pour générer des représentations de mots riches en contexte, qui peuvent ensuite être utilisées pour prédire les étiquettes d'extraction d'information. Par exemple, BERT pourrait apprendre à reconnaître que le mot "Python" dans le contexte d'un CV fait référence à un langage de programmation, et non à un animal.

Les modèles Transformers comme BERT présentent plusieurs avantages pour le parsing de CVs. Ils peuvent capturer des dépendances à long terme dans le texte, et générer des représentations de mots riches en contexte. Cependant, ils nécessitent de grandes quantités de données d'entraînement et de ressources informatiques pour être entraînés efficacement.[35]

3.2.3 Comparaison des Techniques

3.2.3.1 Performance des Approches Traditionnelles vs Modernes

Les approches traditionnelles de parsing de CVs, telles que les techniques basées sur des règles et les modèles statistiques, ont été largement utilisées dans le passé. Cependant, les approches modernes de deep learning, telles que les réseaux de neurones convolutifs (CNNs), les réseaux de neurones récurrents (RNNs) et les modèles Transformers comme BERT, ont montré des performances supérieures dans de nombreuses tâches de traitement du langage naturel, y compris le parsing de CVs.

Selon une étude de Zhang et al. (2020)[35], les modèles de deep learning ont obtenu des résultats significativement meilleurs que les approches traditionnelles dans l'extraction d'informations à partir de CVs. En particulier, les modèles Transformers comme BERT ont montré des performances supérieures à celles des RNNs et des CNNs dans cette tâche.

Cependant, il convient de noter que les approches traditionnelles peuvent encore être utiles dans certains cas, en particulier lorsque les données d'entraînement sont limitées ou

lorsque la tâche de parsing est relativement simple.

3.2.3.2 Cas d'Utilisation Spécifiques pour Chaque Technique

Les différentes techniques de parsing de CVs peuvent être plus adaptées à certains cas d'utilisation que d'autres. Par exemple, les techniques basées sur des règles peuvent être utiles lorsque la structure du CV est relativement simple et cohérente, car elles permettent de définir des règles d'extraction spécifiques pour chaque champ d'information.

Les modèles statistiques, tels que les champs aléatoires conditionnels (CRFs), peuvent être utiles lorsque la structure du CV est plus complexe et que les champs d'information sont étroitement liés les uns aux autres. Les CRFs peuvent prendre en compte les dépendances entre les champs d'information lors de l'extraction, ce qui peut améliorer la précision des résultats.

Les modèles de deep learning, tels que les RNNs, les CNNs et les modèles Transformers comme BERT, peuvent être utiles lorsque les données d'entraînement sont abondantes et que la tâche de parsing est relativement complexe. Ces modèles peuvent apprendre des représentations riches et complexes des données d'entrée, ce qui peut améliorer la précision et la généralisation des résultats d'extraction.

3.2.3.3 Critères de Choix d'une Méthode

Le choix d'une méthode de parsing de CVs dépend de plusieurs facteurs, notamment :

- **La complexité de la tâche de parsing :** si la tâche est relativement simple, les approches traditionnelles peuvent être suffisantes. Si la tâche est plus complexe, les approches de deep learning peuvent être plus adaptées.
- **La taille et la qualité des données d'entraînement :** les approches de deep learning nécessitent généralement des données d'entraînement abondantes et de haute qualité pour obtenir des résultats précis. Si les données d'entraînement sont limitées ou de mauvaise qualité, les approches traditionnelles peuvent être plus adaptées.
- **Le temps et les ressources de calcul disponibles :** les approches de deep learning peuvent nécessiter des temps d'entraînement et des ressources de calcul importants, en particulier pour les modèles complexes comme BERT. Si les temps et les ressources de calcul sont limités, les approches traditionnelles peuvent être plus adaptées.
- **Les exigences de précision et de généralisation :** si la précision et la généralisation des résultats d'extraction sont critiques, les approches de deep learning peuvent être plus adaptées en raison de leur capacité à apprendre des représentations riches et complexes des données d'entrée.

Dans l'ensemble, le choix d'une méthode de parsing de CVs doit être basé sur une évaluation approfondie des avantages et des inconvénients de chaque approche en fonction des exigences et des contraintes spécifiques de la tâche

3.2.3.4 Comparaison avec les Méthodes Existantes

Il existe plusieurs méthodes existantes pour le parsing de CVs, notamment les méthodes basées sur des règles et des modèles statistiques. Les méthodes basées sur des règles impliquent la définition manuelle de règles pour extraire les informations des CVs, tandis que les modèles statistiques impliquent l'utilisation d'algorithmes d'apprentissage automatique pour apprendre des modèles à partir de données annotées.

Les modèles de deep learning ont montré des performances supérieures à celles des méthodes existantes dans le parsing de CVs. Par exemple, Zhang et al. (2020) qui ont proposé un modèle de deep learning basé sur BERT et CRF pour le parsing de CVs chinois, et ont obtenu des performances supérieures à celles des méthodes basées sur des règles et des modèles statistiques.

Cependant, il est important de noter que la performance d'un modèle de parsing de CVs dépend de la qualité et de la diversité des données d'entraînement, ainsi que de l'adéquation de la méthode choisie à la tâche spécifique de parsing de CVs.

3.3 Défis et Limitations dans le Parsing de CVs

Le parsing de CVs est une tâche complexe qui comporte de nombreux défis et limitations. Dans cette section, nous allons discuter de certains des principaux défis et limitations associés au parsing de CVs.

3.3.1 Gestion des Formats Variés de CVs

L'un des principaux défis dans le parsing de CVs est la variété des formats de CVs. Les CVs peuvent être structurés de différentes manières, ce qui rend difficile l'extraction d'informations cohérentes et précises. Les CVs peuvent être au format texte brut, PDF, HTML ou même image. De plus, les CVs peuvent contenir des tableaux, des graphiques et d'autres éléments visuels qui peuvent rendre difficile l'extraction d'informations.

Pour surmonter ce défi, il est important d'utiliser des techniques de prétraitement pour normaliser les CVs et les convertir en un format plus facile à traiter. Les techniques de prétraitement peuvent inclure la conversion de PDF en texte brut, l'extraction de texte à partir d'images et la suppression de tableaux et d'autres éléments visuels

3.3.2 Problèmes de Biais dans les Données d'Entraînement

Un autre défi important dans le parsing de CVs est le biais dans les données d'entraînement. Les données d'entraînement peuvent contenir des biais implicites ou explicites qui peuvent affecter les performances du modèle. Par exemple, si les données d'entraînement

sont principalement composées de CVs d'hommes, le modèle peut être biaisé en faveur des hommes et ne pas fonctionner aussi bien pour les CVs de femmes.

Pour surmonter ce défi, il est important d'utiliser des ensembles de données d'entraînement diversifiés et représentatifs. Il est également important de tester les modèles sur des ensembles de données diversifiés pour s'assurer qu'ils fonctionnent bien pour tous les groupes démographiques.

3.3.3 Limitations Techniques et Exigences en Ressources Computationnelles

Le parsing de CVs peut être une tâche très exigeante en ressources computationnelles, en particulier lors de l'utilisation de modèles de deep learning. Les modèles de deep learning nécessitent souvent de grandes quantités de données d'entraînement et peuvent prendre beaucoup de temps à entraîner. De plus, les modèles de deep learning peuvent être difficiles à interpréter, ce qui peut rendre difficile l'identification des erreurs et des biais.

Pour surmonter ces limitations, il est important d'utiliser des techniques d'optimisation pour réduire les temps d'entraînement et d'utiliser des techniques d'interprétabilité pour comprendre comment les modèles prennent leurs décisions. Il est également important de considérer des modèles plus simples et plus efficaces, tels que les modèles basés sur des règles ou des modèles statistiques, lorsque cela est approprié

3.4 Modèles de Vision par Ordinateur

3.4.1 Introduction à la Computer Vision

Ce chapitre introduit les concepts fondamentaux de la vision par ordinateur, en définissant ses principes de base et ses principales tâches. Les objectifs fondamentaux de la vision par ordinateur sont variés et incluent, entre autres, la reconnaissance et la classification d'objets, la segmentation d'images, la reconstruction en trois dimensions, la détection de mouvements, l'analyse de scènes, la surveillance et la robotique [36].

3.4.2 Détection d'objets

La détection d'objets est une tâche fondamentale de la vision par ordinateur qui consiste à identifier la présence et la localisation d'objets d'intérêt dans une image ou une vidéo. Cette technique repose sur le principe de l'analyse du contenu visuel pour détecter et localiser les différents objets présents dans une scène.

Le processus de détection d'objets s'articule généralement en trois étapes principales :

1. **Extraction des caractéristiques (features) visuelles :** Dans un premier temps, des

descripteurs visuels tels que les formes, les textures, les couleurs ou les contours sont extraits de l'image. Ces caractéristiques permettent de représenter les propriétés distinctives des différents objets. Par exemple, pour la détection de voitures, on pourrait extraire des descripteurs de formes rectangulaires, de couleurs caractéristiques des véhicules (gris, noir, blanc, etc.) et de contours délimitant les différentes parties d'un véhicule (capot, portières, etc.).

2. **Classification des régions candidates :** Ensuite, l'image est divisée en un ensemble de régions potentiellement intéressantes, appelées régions candidates. Un classifieur est alors entraîné pour identifier, parmi ces régions, celles qui contiennent effectivement un objet d'intérêt. Ce classifieur utilise les caractéristiques extraites précédemment pour décider si une région donnée contient ou non un objet cible, comme une voiture par exemple.
3. **Localisation des objets :** Enfin, les boîtes englobantes (bounding boxes) délimitant les objets détectés sont affinées et positionnées avec précision dans l'image. Cette étape permet de définir les coordonnées exactes de chaque objet détecté dans la scène.

Un exemple illustratif de détection d'objets pourrait être la détection de voitures dans une image de scène routière. Les étapes seraient les suivantes :

1. **Extraction des caractéristiques visuelles :** Détection des contours rectangulaires, des zones de couleurs caractéristiques des véhicules (gris, noir, blanc, etc.) et d'autres descripteurs visuels spécifiques aux voitures.
2. **Classification des régions candidates :** Identification des régions de l'image susceptibles de contenir des voitures, à l'aide d'un classifieur entraîné sur des données de voitures. Ce classifieur pourrait par exemple utiliser un réseau de neurones convolutifs (CNN) pour effectuer cette tâche de classification.
3. **Localisation des objets :** Ajustement précis des boîtes englobantes autour de chaque voiture détectée, en affinant les coordonnées des rectangles pour correspondre au mieux aux contours des véhicules.

Cette approche générale de détection d'objets a été formalisée dans un article scientifique fondateur, "Viola-Jones object detection framework" [37], publié en 2001. Cet article a posé les bases des techniques de détection d'objets, en introduisant notamment l'utilisation de caractéristiques visuelles "Haar-like" et d'un classifieur AdaBoost, ouvrant ainsi la voie aux développements ultérieurs dans ce domaine. Les avancées récentes en apprentissage profond (deep learning) ont ensuite permis de nouvelles avancées significatives dans la détection d'objets, avec des architectures telles que R-CNN, Faster R-CNN et YOLO.

3.4.3 Segmentation d'instances

La segmentation d'instances est une tâche avancée de vision par ordinateur qui consiste à identifier de manière précise les différents objets présents dans une image, en allant au-delà de la simple détection d'objets. Contrairement à la détection d'objets qui se concentre sur la délimitation des objets à l'aide de boîtes englobantes, la segmentation d'instances vise à en déterminer les contours exacts, pixel par pixel.

Ce processus s'appuie sur des techniques d'apprentissage profond (*deep learning*) qui permettent de segmenter finement chaque instance d'un même type d'objet (par exemple, chaque personne dans une foule) au sein de la scène. La segmentation d'instances est généralement réalisée à l'aide d'architectures de réseaux de neurones convolutifs (CNN) sophistiquées, telles que Mask R-CNN [38].

Le fonctionnement de ces modèles de segmentation d'instances peut être décomposé en plusieurs étapes clés :

- **Détection d'objets :** La première étape consiste à détecter la présence et la localisation des objets d'intérêt dans l'image, à l'aide d'un module de détection d'objets intégré au modèle.
- **Extraction de caractéristiques visuelles :** Ensuite, un encodeur CNN extrait des descripteurs visuels riches à partir de l'image d'entrée, capturant les formes, les textures et les motifs caractéristiques des objets détectés.
- **Segmentation sémantique :** Un module de segmentation sémantique est alors appliqué pour associer à chaque pixel de l'image une étiquette sémantique correspondant à l'objet auquel il appartient (personne, voiture, arbre, etc.).
- **Individualisation des instances :** Enfin, un module de segmentation d'instances utilise ces informations sémantiques pour délimiter précisément les contours de chaque instance d'objet, même lorsqu'ils se touchent ou se chevauchent partiellement.

Un exemple illustratif de segmentation d'instances pourrait être l'identification précise de chaque personne dans une image de foule. Contrairement à la simple détection de personnes, la segmentation d'instances permettrait de délimiter les contours exacts de chaque individu, en séparant clairement chaque instance, même lorsque les personnes se trouvent côte à côte ou partiellement occultées.

Les progrès continus dans ce domaine, notamment grâce à l'apprentissage profond, ouvrent la voie à de nombreuses applications innovantes, en offrant une compréhension toujours plus fine de l'environnement visuel, que ce soit pour la robotique, l'analyse d'images médicales ou encore la réalité augmentée.

3.4.4 Présentation des modèles Faster R-CNN et Mask R-CNN

L'évolution des techniques de détection d'objets dans le domaine de la vision par ordinateur a été marquée par des avancées significatives avec l'introduction des architectures basées sur les réseaux de neurones convolutifs (CNN). Parmi celles-ci, les modèles R-CNN (Region-based Convolutional Neural Networks) et ses successeurs ont joué un rôle crucial. Pour comprendre pleinement les modèles les plus sophistiqués tels que Faster R-CNN et Mask R-CNN, il est essentiel de revisiter d'abord les concepts fondamentaux et les améliorations introduites par les modèles R-CNN et Fast R-CNN.

Les R-CNN ont révolutionné la détection d'objets en combinant les capacités de représentation visuelle des CNN avec une approche de région d'intérêt. Cependant, malgré leur précision, ils ont souffert de limitations en termes de temps de calcul et d'efficacité. Fast R-CNN a apporté des solutions clés à ces problèmes, améliorant ainsi la rapidité et la performance globale du modèle. En examinant en profondeur ces deux architectures, nous

serons mieux équipés pour apprécier les innovations apportées par Faster R-CNN et Mask R-CNN.

3.4.5 R-CNN et Fast R-CNN

3.4.5.1 R-CNN (Region-based Convolutional Neural Networks)

L'architecture R-CNN, introduite en 2014 dans l'article scientifique "Rich feature hierarchies for accurate object detection and semantic segmentation" par Girshick et al. [39] repose sur l'utilisation de réseaux de neurones convolutifs (CNN) pour effectuer la détection d'objets en 4 étapes principales :

— **Génération de régions candidates :** L'objectif de cette première étape est d'identifier dans l'image les zones susceptibles de contenir des objets d'intérêt.

Pour cela, l'algorithme de recherche sélective (Selective Search) est utilisé. Cet algorithme part d'une sur-segmentation initiale de l'image en petites régions, puis combine récursivement ces régions en utilisant des critères de similarité tels que la couleur, la texture, la taille, etc.

Le résultat est une liste d'environ 2 000 régions candidates, appelées "boîtes englobantes", qui couvrent de manière assez exhaustive les objets potentiels dans l'image.

— **Extraction de caractéristiques CNN :**Chacune des 2 000 régions candidates est ensuite redimensionnée à une taille fixe (par exemple 227x227 pixels) pour pouvoir être utilisée comme entrée à un réseau de neurones convolutifs (CNN) pré-entraîné.

Dans l'article, les auteurs ont utilisé le modèle AlexNet, qui était à l'époque l'un des réseaux CNN les plus performants pour la classification d'images.

Le CNN est alors appliqué à chaque région candidate indépendamment, permettant d'extraire un vecteur de caractéristiques de dimension 4096 pour chacune d'entre elles.

— **Classification et régression des boîtes englobantes :** Les vecteurs de caractéristiques CNN sont ensuite utilisés comme entrée à un modèle SVM (Support Vector Machine) binaire, entraîné indépendamment pour chaque classe d'objet.

Le SVM produit un score de confiance pour la présence de l'objet de chaque classe dans chaque région candidate.

En parallèle, un modèle de régression linéaire est entraîné pour affiner la localisation de la boîte englobante de chaque objet détecté, en prédisant les coordonnées (x, y, largeur, hauteur) de la boîte.

— **Post-traitement :** Après cette étape de classification et de régression, on dispose d'un ensemble de boîtes englobantes avec leurs scores de confiance associés pour chaque classe d'objet.

Un seuillage est appliqué pour éliminer les détections les plus faibles, en ne conservant que celles ayant un score supérieur à un certain seuil (par exemple 0.5).

Enfin, un algorithme de suppression non-maximale (Non-Maximum Suppression) est utilisé pour éliminer les détections redondantes correspondant au même objet. Cet algorithme conserve uniquement la détection ayant le score le plus élevé parmi les boîtes ayant un recouvrement spatial trop important.

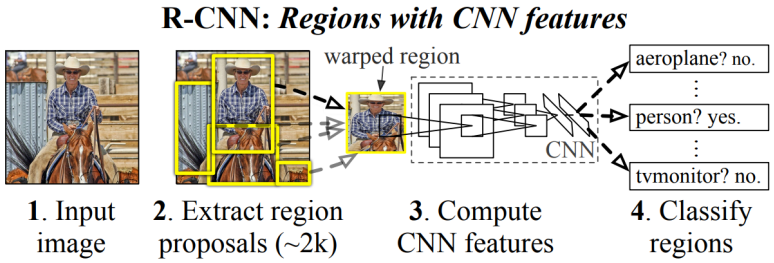


FIGURE 3.1 – R-CNN : régions dotées de fonctionnalités CNN [39]

Bien que l'approche R-CNN ait marqué un tournant dans la détection d'objets, elle présentait également certaines limitations, notamment en termes de temps de calcul et de complexité du pipeline. En effet, le processus d'inférence impliquait de redimensionner chaque région candidate, de la transmettre au réseau CNN et d'effectuer les calculs de classification et de régression pour chacune d'entre elles, ce qui était extrêmement coûteux en ressources.

Ces défis ont par la suite été adressés par des variantes telles que Fast R-CNN

3.4.5.2 Fast R-CNN

Bien que l'approche R-CNN ait apporté des améliorations significatives par rapport aux méthodes de détection d'objets existantes, elle souffrait de certaines limitations en termes de temps de calcul et de complexité du pipeline. Pour remédier à ces problèmes, Girshick a proposé en 2015 une nouvelle architecture appelée Fast R-CNN[40].

L'approche Fast R-CNN conserve les principes clés de R-CNN, à savoir l'utilisation de régions candidates générées par un algorithme de type Selective Search, suivie de l'application d'un réseau de neurones convolutifs (CNN) pour l'extraction de caractéristiques visuelles robustes. Cependant, Fast R-CNN introduit des améliorations significatives dans la mise en œuvre :

- **Extraction des caractéristiques CNN sur l'image entière :** Contrairement à R-CNN qui appliquait le réseau CNN de manière indépendante sur chaque région candidate, Fast R-CNN n'applique le CNN qu'une seule fois sur l'image complète.

  Les caractéristiques CNN sont ensuite projetées sur les régions candidates identifiées par l'algorithme de recherche sélective.

  Cela permet de partager les calculs du CNN entre toutes les régions candidates, évitant ainsi les calculs redondants.
- **Classification et régression des boîtes englobantes en une seule étape :** Les branches de classification (prédiction des probabilités de chaque classe) et de régression des boîtes englobantes sont intégrées dans un seul et même réseau CNN.

Cela permet une optimisation conjointe et différentiable de ces deux objectifs, à la différence de R-CNN où ces étapes étaient séparées.

— **Réseau CNN avec couche de "ROI Pooling" :** Le réseau CNN utilisé dans Fast R-CNN a sa couche de pooling finale remplacée par une couche de "ROI Pooling" (Region of Interest Pooling).

Cette couche divise les features issues du CNN en sous-fenêtres de taille fixe (7x7 dans l'article), permettant d'obtenir des features de taille constante quelle que soit la taille de la région candidate.

Cela permet d'alimenter directement les couches fully connected suivantes, contrairement à R-CNN qui devait redimensionner chaque région candidate.

— **Schéma d'entraînement multi-tâche :** L'entraînement de Fast R-CNN se fait de manière différentiable et conjointe, optimisant à la fois la classification et la régression des boîtes englobantes.

Cela permet d'améliorer les performances en tirant parti de l'influence mutuelle entre ces deux tâches.

De plus, l'entraînement fin-tuning des couches convolutionnelles du CNN, en plus des couches fully connected, s'est avéré important pour obtenir de meilleures performances.

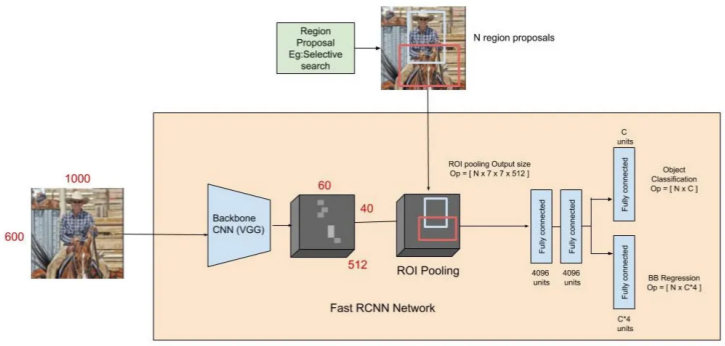


Figure 3.2 – Pipeline Fast R-CNN [41]

Bien que Fast R-CNN ait apporté des améliorations substantielles par rapport à R-CNN, certaines limitations subsistaient, notamment en termes de temps de calcul pour la génération des régions candidates. Pour résoudre ce problème, Ren et al. ont ensuite proposé l'architecture Faster R-CNN, qui intègre également un réseau de proposition de régions dans le pipeline, permettant d'accélérer davantage le processus de détection d'objets.

3.4.6 Faster R-CNN et Mask R-CNN

3.4.6.1 Faster R-CNN

Bien que Fast R-CNN ait apporté des améliorations substantielles par rapport à R-CNN, certaines limitations subsistaient, notamment en termes de temps de calcul pour la génération des régions candidates. Pour résoudre ce problème, Ren et al. ont proposé en 2015 une nouvelle architecture appelée Faster R-CNN [42]..

Principes de l'approche Faster R-CNN :

- **Réseau de proposition de régions (Region Proposal Network - RPN) :** Le RPN est une sous-partie du réseau Faster R-CNN qui se concentre sur la génération de régions d'intérêt (region proposals) potentiellement contenant des objets.

  Il utilise une approche de fenêtre glissante sur les cartes de caractéristiques obtenues en amont par un réseau CNN de base (backbone).

  Pour chaque position de la fenêtre glissante, le RPN génère plusieurs "anchor boxes" de différentes tailles et ratios d'aspect prédéfinis.

  Chaque anchor box est associée à deux sorties :

  - **1 :**Un score d'"objectness" qui représente la probabilité que l'anchor box contienne un objet d'intérêt plutôt que du background.
  - **2 :** Des ajustements des coordonnées de l'anchor box pour mieux correspondre à la taille et à la position réelle de l'objet.

  Le NMS (Non-Maximum Suppression) est ensuite appliqué pour ne conserver que les meilleures régions proposals en éliminant les propositions redondantes.
- **Détecteur Fast R-CNN :** Le détecteur Fast R-CNN prend les régions proposals générées par le RPN et les traite pour détecter et localiser les objets.

  La technique de RoI Pooling est utilisée pour transformer les régions proposals de tailles variables en cartes de caractéristiques de taille fixe.

  Ces cartes de caractéristiques sont ensuite traitées par des couches fully connected pour effectuer deux tâches :

  - **1 :** La classification de l'objet dans l'une des classes prédéfinies (ou background).
  - **2 :** La régression des paramètres de la boîte englobante de l'objet pour affiner sa localisation.

  Une fonction de perte multi-tâche est utilisée, combinant la perte de classification et la perte de régression des boîtes.

  Après la prédiction des probabilités de classes et des ajustements des boîtes, un post-traitement avec NMS est appliqué pour raffiner les résultats finaux de détection.
- **Entraînement end-to-end :**L'entraînement du Faster R-CNN se fait de manière end-to-end, c'est-à-dire que l'ensemble du réseau, incluant les couches ajoutées pour le RPN et les couches convolutives partagées, est optimisé conjointement.

L'approche d'échantillonnage utilisée est "image-centric" : chaque batch d'entraînement est constitué d'anchor boxes provenant d'une seule image, avec un mélange équilibré d'anchor boxes positifs (contenant un objet) et négatifs (background).

Les nouvelles couches ajoutées pour le RPN sont initialisées avec des poids aléatoires, tandis que les couches convolutives existantes (backbone) utilisent des poids pré-entraînés sur ImageNet, suivant les pratiques standard.

Des paramètres d'entraînement appropriés sont utilisés, comme un taux d'apprentissage décroissant, un momentum et un poids de régularisation.

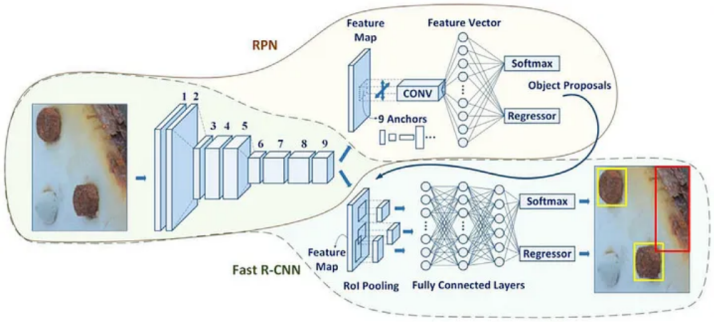


Figure 3.3 – Architecture Faster R-CNN [43]

Grâce à ces innovations, Faster R-CNN a établi de nouveaux standards de performance et de vitesse pour la détection d'objets basée sur des régions.

3.4.6.2 Mask R-CNN

Mask R-CNN est une extension du modèle Faster R-CNN, introduite par Kaiming He et al [38]. Mask R-CNN ajoute une capacité de segmentation d'instance aux tâches de détection et de classification d'objets réalisées par le Faster R-CNN.

Principes de l'approche R-CNN :

- **Segmentation d'instance** : La branche de segmentation d'instance prend les cartes de caractéristiques issues du RoI Pooling et prédit un masque binaire pour chaque objet détecté. Ce masque indique les pixels appartenant à l'objet, permettant ainsi une segmentation fine de l'instance. La prédiction du masque se fait de manière indépendante pour chaque classe d'objet détectée.
- **Alignement des caractéristiques (Feature Alignment)** : Contrairement au RoI Pooling du Faster R-CNN qui utilise une opération de max pooling, Mask R-CNN utilise une technique appelée RoIAlign. RoIAlign évite la quantification des coordonnées des régions d'intérêt, préservant ainsi mieux les informations spatiales nécessaires à une segmentation fine.

— **Fonction de perte multi-tâche** : Mask R-CNN utilise une fonction de perte combinant les pertes de classification, de régression des boîtes englobantes et de segmentation des masques. La perte de segmentation des masques est calculée à l'aide d'une erreur binaire croisée (binary cross-entropy) entre le masque prédit et le masque de vérité terrain.

— **Entraînement end-to-end** : Comme le Faster R-CNN, Mask R-CNN est entraîné de manière end-to-end, optimisant conjointement toutes les branches du réseau. L'initialisation des poids suit également les mêmes principes, avec des poids pré-entraînés pour le backbone CNN et des poids aléatoires pour les nouvelles couches.

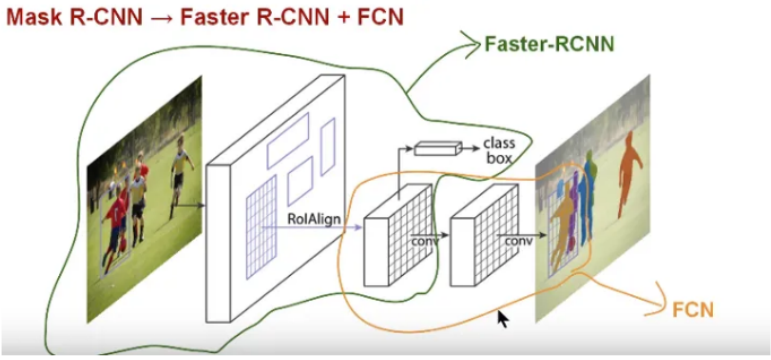


Figure 3.4 – Mask R-CNN Architecture [44]

En résumé, Mask R-CNN étend les capacités du Faster R-CNN en ajoutant une branche de segmentation d'instance, permettant ainsi une détection d'objets plus fine et détaillée. Cette avancée a grandement amélioré les performances des modèles de vision par ordinateur sur des tâches complexes de détection et de segmentation d'objets.

3.5 Modèles pour le Traitement du Langage Naturel

Le traitement du langage naturel (TLN) a considérablement évolué avec l'avènement des modèles de transformateurs et des modèles de langage à grande échelle (Large Language Models, LLMs). Ces algorithmes ont transformé la manière dont les systèmes intelligents comprennent, génèrent et interagissent avec le langage humain.

Les Transformateurs : Une Révolution dans le TLN Les transformateurs, introduits par l'article phare "Attention is All You Need" de Vaswani et al. en 2017, ont révolutionné le TLN en éliminant les limitations des architectures séquentielles comme les Réseaux de Neurones Récurrents (RNN). Les transformateurs utilisent un mécanisme d'attention auto-régressive qui permet de traiter simultanément toutes les positions d'une séquence, facilitant ainsi l'apprentissage des dépendances à long terme et réduisant significativement le temps de calcul grâce au parallélisme.

3.5.1 CamemBERT

CamemBERT est un modèle de langage basé sur l'architecture BERT (Bidirectional Encoder Representations from Transformers), spécialement préentraîné sur un large corpus de textes en français. Développé par les chercheurs de l'INRIA et de Facebook AI[45], CamemBERT est optimisé pour le traitement des textes en français et se distingue par sa capacité à comprendre les nuances et les contextes propres à cette langue.

3.5.2 Comprendre la Modélisation du Français par CamemBERT

Pour télécharger le modèle CamemBERT grâce à la librairie HuggingFace, il suffit d'une ligne de code :

```
from transformers import CamembertForMaskedLM

camembert = CamembertForMaskedLM.from_pretrained('camembert-base')
```

Figure 3.5 – Télécharger le modèle CamemBERT

HuggingFace est une bibliothèque en ligne gratuite de modèles pré-entraînés et de jeux de données (datasets) qui permet d'utiliser les architectures et les poids des modèles mis à disposition par les équipes de recherche et les institutions.

3.5.2.1 L'Architecture de CamemBERT : Le Transformer

Le Transformer est une architecture deep learning introduite en 2017 dans l'article Attention Is All You Need. Il permet de traiter des séquences dont les éléments sont fortement inter-dépendants, comme c'est le cas pour les mots d'une phrase.

CamemBERT-base (la version de base du modèle) est composé de plusieurs éléments :

- Couche d'Embedding : Représente chaque mot en vecteur, transformant les mots en une forme numérique compréhensible par le modèle.
- 12 Couches Cachées : Composées principalement de deux types de transformations :
  - Self-Attention : Permet au modèle de se concentrer sur différentes parties de la phrase pour comprendre le contexte de chaque mot.
  - Transformations Denses : Appliquent des transformations linéaires suivies de fonctions d'activation, aidant à capturer les relations complexes entre les mots.

Schématiquement :

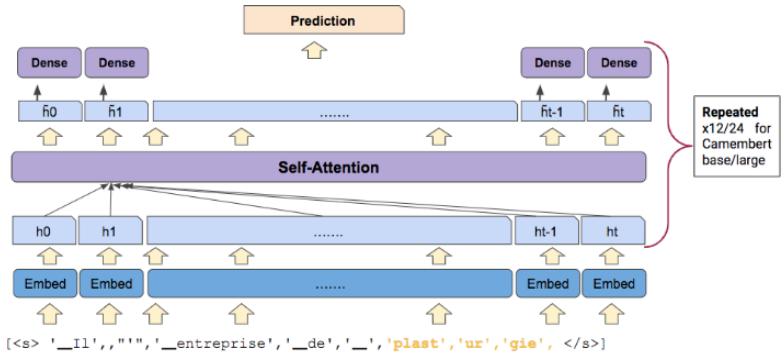


FIGURE 3.6 – Architecture du modèle CamemBERT [46]

3.5.2.2 Préentraînement sur des Corpus Français

Contrairement aux modèles multilingues, CamemBERT est entraîné exclusivement sur des textes français, ce qui lui permet de saisir les subtilités et les particularités linguistiques du français mieux que les modèles généralistes. Les corpus utilisés incluent des sources variées telles que des articles de presse, des œuvres littéraires, et des discussions sur les forums en ligne, couvrant ainsi un large éventail de registres et de styles de langue.

3.5.2.3 Applications et Avantages pour le Français

CamemBERT est utilisé pour diverses tâches de NLP telles que la classification de texte, l'extraction d'entités nommées, l'analyse de sentiment, et la génération de texte. Dans le cadre de l'analyse des CV, il peut être utilisé pour extraire des informations clés comme les compétences, les expériences professionnelles et les qualifications.

En étant spécifiquement optimisé pour le français, CamemBERT offre une précision et une performance accrues pour les tâches de NLP en français par rapport aux modèles multilingues ou non spécialisés. Cette spécialisation permet d'éviter les erreurs courantes des modèles multilingues, telles que les ambiguïtés et les incompréhensions des expressions idiomatiques ou des constructions syntaxiques complexes propres au français.

3.5.2.4 Défis et Solutions

L'une des difficultés majeures dans l'utilisation de CamemBERT pour la NER dans les CV est la variabilité du langage utilisé. Les candidats peuvent employer des termes différents pour décrire des compétences similaires, ou utiliser des abréviations et des jargons spécifiques à leur domaine. Pour surmonter ce défi, il est possible de combiner CamemBERT avec des techniques de normalisation des textes et de construire des dictionnaires de synonymes spécifiques au domaine des CV.

3.5.2.5 Déploiement de CamemBERT

CamemBERT, étant optimisé pour les textes en français, peut être déployé sur des serveurs équipés de GPU de moyenne gamme pour des performances maximales. L'intégration de CamemBERT dans une application de traitement de CV nécessite la préparation des textes en entrée, leur passage par le modèle pour l'extraction d'entités nommées, et le stockage des résultats dans une base de données. Les environnements de déploiement doivent être configurés pour gérer efficacement les requêtes et les volumes de données spécifiques aux CV en français. Par exemple, une architecture de microservices peut être utilisée pour scalabiliser le traitement et réduire la latence des requêtes.

3.5.3 Modèle LLM (Large Language Models)

Les modèles de grande envergure, ou Large Language Models (LLM), représentent une avancée majeure dans le domaine du traitement automatique du langage naturel. Ces modèles, souvent basés sur l'architecture Transformer, sont entraînés sur d'énormes corpus de données textuelles, leur permettant d'apprendre des représentations complexes du langage. Pour notre projet d'analyse de CV, nous avons choisi d'utiliser Mistral 7B, un modèle LLM performant.

3.5.4 Mistral 7B

Mistral 7B est un modèle de langage de 7,3 milliards de paramètres développé par Mistral AI. Il est conçu pour surpasser d'autres modèles de taille similaire, et même plus grands, dans divers benchmarks. Par exemple, il dépasse le modèle Llama 2 de 13 milliards de paramètres sur toutes les tâches et surpasse le modèle Llama 1 de 34 milliards de paramètres sur de nombreux benchmarks. Il bénéficie d'un entraînement approfondi sur des corpus textuels étendus. Ce modèle se distingue par sa capacité à interpréter et générer du texte avec une grande précision, offrant des applications variées et des résultats d'une qualité remarquable.

3.5.4.1 L'Architecture de Mistral 7B

Comme de nombreux modèles LLM modernes, Mistral 7B est construit sur l'architecture Transformer. Comparé à des modèles plus petits comme CamemBERT, Mistral 7B, avec ses 7 milliards de paramètres, capte des relations contextuelles d'une complexité accrue. Cela se traduit par une compréhension plus profonde et une génération plus fluide du texte, rendant le modèle particulièrement puissant pour des tâches de TALN sophistiquées.

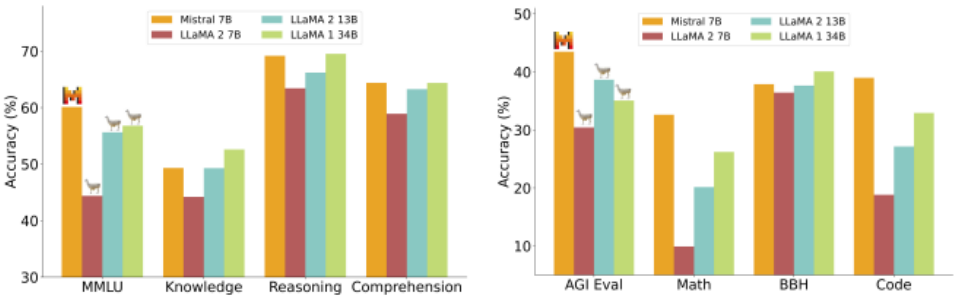


Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior to Llama 1 34B in mathematics, code generation, and reasoning benchmarks.

Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.2%**

Table 2: Comparison of Mistral 7B with Llama. Mistral 7B outperforms Llama 2 13B on all metrics, and approaches the code performance of Code-Llama 7B without sacrificing performance on non-code benchmarks.

FIGURE 3.7 – Performance du modèle Mistral 7B
[47]

3.5.4.2 Préentraînement sur des Corpus Diversifiés

Mistral 7B a été préentraîné sur une large gamme de corpus textuels, incluant différentes langues et domaines. Cette diversité lui confère une polyvalence exceptionnelle, lui permettant de traiter des textes issus de multiples contextes et de comprendre des nuances linguistiques variées. Cette robustesse en fait un outil idéal pour des applications généralisées et spécifiques.

3.5.4.3 Modélisation de Mistral 7B pour le Fine-Tuning

La force de Mistral 7B réside dans sa flexibilité. Grâce à sa taille et à son architecture, il est particulièrement bien adapté au fine-tuning, un processus qui consiste à adapter le modèle pré-entraîné à des tâches spécifiques en utilisant des données supplémentaires pertinentes. Dans le cadre de notre projet, nous avons fine-tuné Mistral 7B sur un corpus de CV annotés, optimisant ainsi sa capacité à reconnaître et extraire des informations précises.

3.5.4.4 Applications et Avantages pour le Français et la Reconnaissance d'Entités Nommées (NER)

Grâce à sa puissance et sa flexibilité, Mistral 7B excelle dans de nombreuses applications de NLP telles que la traduction automatique, la génération de texte, et l'analyse de

sentiment. Pour notre projet d'analyse de CV, nous nous sommes concentrés sur la reconnaissance d'entités nommées (NER). Le fine-tuning de Mistral 7B sur notre corpus de CV a permis d'améliorer considérablement la précision de l'extraction d'informations, facilitant la gestion efficace et précise des candidatures.

3.5.4.5 Défis et Solutions

L'utilisation de Mistral 7B pose plusieurs défis, notamment en termes de ressources de calcul et de gestion des coûts d'infrastructure. Le modèle, avec ses 7 milliards de paramètres, nécessite des serveurs équipés de GPU performants pour fonctionner efficacement. La gestion de la latence et l'optimisation des performances sont cruciales pour assurer un déploiement réussi, surtout pour des applications en temps réel comme l'analyse de CV. Pour surmonter ces défis, nous avons mis en place des stratégies d'optimisation spécifiques et choisi une infrastructure adaptée à nos besoins.

3.5.4.6 Déploiement de Mistral 7B

Le déploiement de Mistral 7B nécessite une infrastructure robuste capable de gérer des charges de travail intensives. Pour intégrer ce modèle dans notre système d'analyse de CV, nous l'avons déployé en tant que service backend, accessible via une API FastAPI et une quantification de poids. Cela permet de traiter de grandes quantités de texte de manière rapide et efficace. Une attention particulière a été portée à l'optimisation des performances et à la gestion des coûts d'infrastructure pour garantir une utilisation

3.5.4.7 Synthèse

Pour notre projet d'analyse de CV, CamemBERT et Mistral 7B offrent des avantages distincts mais complémentaires. CamemBERT, avec son entraînement spécifique sur des textes en français, est idéal pour les tâches nécessitant une compréhension fine et nuancée du langage français. Sa capacité à saisir les particularités linguistiques du français permet une extraction précise et fiable des informations dans les CV rédigés dans cette langue.

D'un autre côté, Mistral 7B, avec ses 7 milliards de paramètres et son entraînement sur des corpus diversifiés et de grande envergure, apporte une flexibilité et une puissance exceptionnelles pour traiter une large gamme de tâches de NLP. Sa capacité à capturer des relations contextuelles complexes et sa polyvalence lui permettent de gérer efficacement des textes dans différents contextes et domaines.

En combinant CamemBERT et Mistral 7B, nous pouvons tirer parti de leurs forces respectives : CamemBERT assure une précision élevée pour les textes en français grâce à son entraînement spécifique, tandis que Mistral 7B apporte une robustesse et une capacité d'adaptation accrues pour des tâches de NLP plus générales et diversifiées. Cette combinaison permet d'améliorer la précision et l'efficacité de l'extraction et de l'analyse des informations contenues dans les CV, offrant ainsi une solution robuste, précise et adaptable

Caractéristiques	CamemBERT	Mistral 7B
Spécialisation	Textes en français	Multilingue et contextes variés
Paramètres	Modèle de taille moyenne (moins de 1 milliard)	7,3 milliards
Applications	NLP en français, extraction de CV	Traduction, génération de texte, reconnaissance d'entités nommées (NER)
Entraînement	Textes exclusivement en français	Corpus diversifiés
Défis	Variabilité du langage des CV	Ressources de calcul nécessaires, coûts d'infrastructure
Déploiement	Serveurs GPU de moyenne gamme	Infrastructure robuste, accessible via API

Table 3.1 – Comparaison entre CamemBERT et Mistral 7B

aux besoins variés de chaque client.

3.6 OCR (Reconnaissance Optique de Caractères)

3.6.1 Introduction à l'OCR

La reconnaissance optique de caractères (OCR) est une technologie qui convertit les textes contenus dans des images ou des documents scannés en données numériques exploitables. Cette conversion permet de transformer des documents physiques ou des images de texte en fichiers éditables et consultables, facilitant ainsi le traitement, la gestion et l'analyse des informations textuelles.

Dans le contexte de notre projet, l'OCR joue un rôle essentiel après que les modèles Faster R-CNN ont détecté les différentes zones d'intérêt dans les CV, telles que l'éducation, l'expérience, les compétences, les informations personnelles, et le nom. Une fois ces zones identifiées, l'OCR est utilisé pour extraire le texte contenu dans ces sections, ce qui permet d'analyser et de structurer les informations de manière automatique.

3.6.2 Définition et importance de l'OCR dans le processus de traitement de documents

L'OCR, ou Reconnaissance Optique de Caractères, est une technologie qui permet de convertir des images de texte manuscrit ou imprimé en texte lisible par machine. Cela inclut des documents numérisés, des photos de documents, des inscriptions sur des images ou des vidéos, et tout autre média contenant du texte sous forme graphique. Le processus OCR détecte les caractères individuels dans l'image, les reconnaît et les transcrit en format texte

numérique.

Importance de l'OCR

- **Automatisation de la Saisie de Données :** L'OCR automatise la transformation des informations textuelles en format numérique, réduisant ainsi le besoin de saisie manuelle de données. Dans le contexte du traitement de CV, cela permet de rapidement extraire et analyser les informations clés sans intervention humaine, améliorant l'efficacité et réduisant les erreurs de transcription.
- **Accélération du Flux de Travail :** En convertissant les documents papier ou les images en formats éditables, l'OCR accélère le processus de traitement des documents. Cela est particulièrement bénéfique pour les entreprises qui traitent de grands volumes de CV, facilitant le tri, la recherche et l'analyse des informations.
- **Accessibilité et Utilisation de Données :** Les documents numérisés ou les images contenant du texte ne sont pas facilement consultables ou modifiables. L'OCR rend ces documents accessibles en extrayant le texte, permettant ainsi la recherche de mots-clés, l'édition, et l'analyse des données. Dans le traitement des CV, cela permet une extraction et une structuration automatiques des informations telles que les compétences, les expériences et les qualifications.
- **Intégration avec d'Autres Systèmes :** Les textes extraits via OCR peuvent être intégrés dans des bases de données, des systèmes de gestion de contenu, ou des plateformes de recrutement. Cela facilite la gestion des informations des candidats et l'automatisation des processus de sélection.
- **Amélioration de la Précision de l'Extraction de Données :** Avec l'OCR, même les documents anciens ou manuscrits peuvent être convertis en données numériques précises. Pour les CV, qui peuvent varier considérablement en format et en style, l'OCR assure que les informations critiques ne sont pas perdues lors de la numérisation.

En somme, l'OCR est une technologie incontournable pour l'automatisation du traitement de documents, en particulier pour les tâches nécessitant l'extraction rapide et précise de textes à partir d'images. Pour notre projet, l'utilisation de l'OCR après la détection des zones clés des CV par Faster R-CNN permet de structurer efficacement les informations et d'améliorer le flux de travail global. De surtout pouvoir transformer notre data d'une data image vers une data textuelle.

3.7 Conclusion

En conclusion, le parsing de CVs est un domaine en constante évolution, avec des défis et des limitations à relever. Les approches traditionnelles et modernes ont toutes deux leurs avantages et leurs inconvénients, et le choix de la méthode dépend des besoins et des contraintes spécifiques du projet.

Les défis dans le parsing de CVs comprennent la gestion des formats variés de CVs, les problèmes de biais dans les données d'entraînement, et les limitations techniques et les exigences en ressources computationnelles. Pour surmonter ces défis, il est important de se tenir informé des dernières avancées dans le domaine, d'utiliser des ensembles de données

diversifiés et représentatifs, et d'optimiser les modèles pour réduire les temps d'entraînement et les besoins en ressources.

Notre travail durant le stage à consister de mettre en pratique les connaissances acquises dans ce chapitre, en développant une solution de parsing de CVs adaptée aux besoins de l'entreprise d'accueil. Nous devrons prendre en compte les défis et les limitations du domaine, et choisir la méthode la plus appropriée pour notre projet.

Chapitre 4

Optimisation par Fine-Tuning des Modèles de Vision par Ordinateur pour la détection de zones textuelles : Approches et Contributions

4.1 Introduction

La détection de zones textuelles dans les documents visuels, comme les CV, constitue un défi essentiel pour de nombreuses applications en vision par ordinateur. Les algorithmes avancés de détection d'objets, notamment Faster R-CNN et Mask R-CNN, ont montré leur efficacité remarquable dans divers contextes. Ce chapitre se penche sur l'adaptation et l'optimisation de ces modèles pour la tâche spécifique de la détection de sections textuelles dans les CV. Nous mettons en avant les techniques de fine-tuning pour ajuster les modèles pré-entraînés à cette application précise, en détaillant les modifications nécessaires au niveau des architectures et des paramètres d'entraînement pour améliorer la précision et la robustesse de la détection textuelle dans des documents visuels structurés.

4.2 Fine-tuning des modèles Faster R-CNN et Mask R-CNN a la tache de detection des zones textuelles dans les CVs

Le fine-tuning des modèles de détection d'objets comme Faster R-CNN et Mask R-CNN est crucial pour les adapter à des tâches spécifiques telles que la détection de sections textuelles dans les CV. Ce processus implique le réentraînement partiel de ces réseaux sur des jeux de données spécifiques, en ajustant finement les couches intermédiaires et su-

périeures pour capter les particularités des textes. Les étapes clés incluent la préparation et l'annotation minutieuse des jeux de données de CV, l'ajustement des hyperparamètres comme les taux d'apprentissage et les seuils de détection, et l'application de techniques de transfert d'apprentissage en utilisant des poids pré-entraînés. L'optimisation des couches intermédiaires permet de préserver les caractéristiques générales tout en accentuant celles spécifiques aux textes des CV. De plus, l'utilisation de stratégies de régularisation et de techniques d'augmentation des données aide à améliorer la robustesse et à prévenir le sur-apprentissage. En combinant ces techniques, on peut considérablement améliorer les performances des modèles pour la détection précise des sections textuelles dans les CV.

4.2.1 Préparation des données

4.2.1.1 Collecte des données

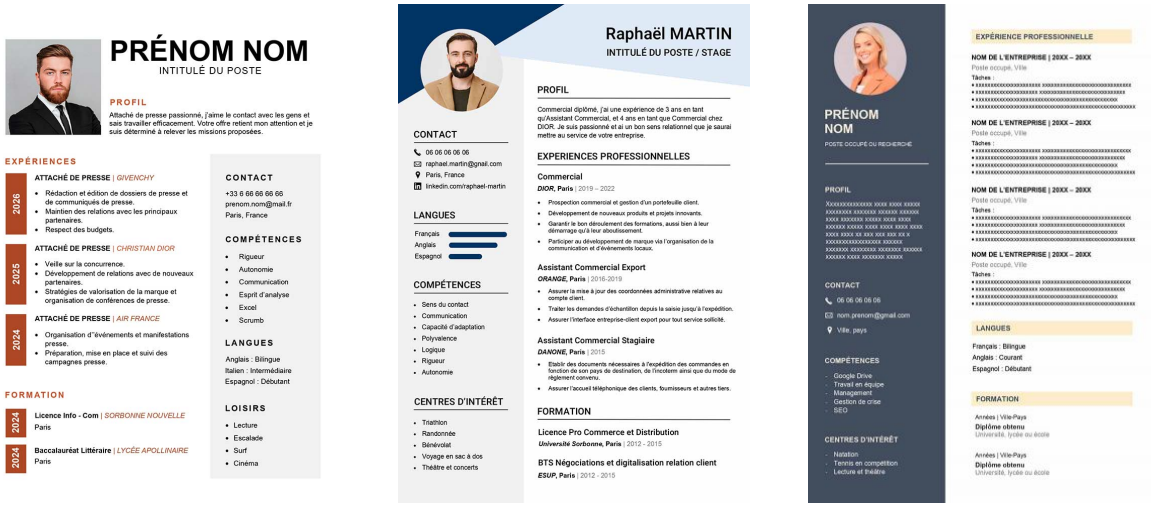
Dans le cadre de cette étude de cas, la collecte et la préparation des données ont été optimisées grâce à l'accès aux vastes ressources internes de l'entreprise hôte. Cette entreprise possède une base de données volumineuse de CV numérisés, accumulée au cours des nombreux processus de recrutement menés au fil des ans. Cet accès privilégié a permis d'amasser un jeu de données diversifié et représentatif, intégrant une large variété de styles et de formats de CV.

Diversité des CV

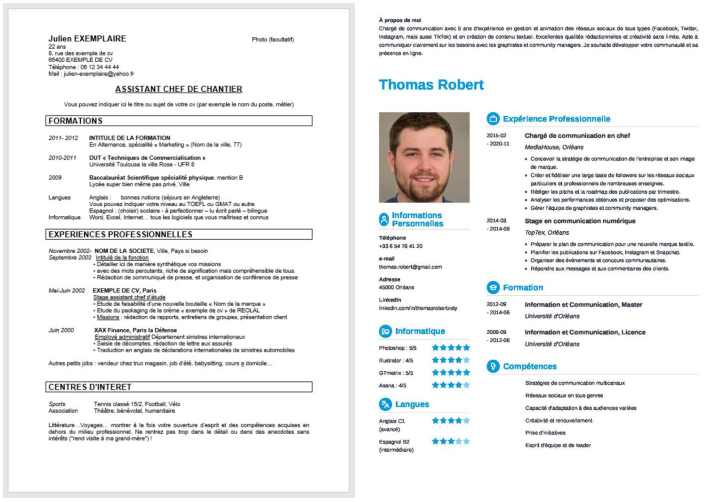
Lors de la collecte, une attention méticuleuse a été portée à l'inclusion de CV provenant de divers contextes géographiques et professionnels. Par exemple, les CV collectés incluent des candidats de différents continents, ce qui reflète une large gamme de pratiques culturelles en matière de présentation de CV. Cette diversité est essentielle pour entraîner un modèle capable de traiter des CV de diverses origines dans le cadre des processus de recrutement de l'entreprise. Par exemple, les CV d'Europe de l'Est ont souvent des formats stricts et détaillés, tandis que ceux des États-Unis peuvent être plus variés et centrés sur les accomplissements personnels.

Hétérogénéité des Formats

L'analyse des CV collectés a révélé une hétérogénéité significative dans la mise en forme des différentes sections. Certains CV suivent un format structuré avec des sections clairement délimitées, telles que "Expérience Professionnelle", "Formation", et "Compétences". À l'inverse, d'autres CV adoptent une structure plus libre, où les informations sont organisées sous forme de blocs de texte. Cette disparité pose un défi particulier pour les modèles de détection d'objets. Par exemple, les CV de professionnels créatifs, comme les designers graphiques, ont souvent des mises en page innovantes et non conventionnelles, tandis que les CV de secteurs plus traditionnels, comme la finance, tendent à suivre des formats plus standardisés.



(a) Image 1 (b) Image 2 (c) Image 3



(d) Image 3 (e) Image 3

FIGURE 4.1 – Différentes mises en page de curriculum vitae

4.2.2 Prétraitement des données

Avant de procéder à l'annotation manuelle des différentes sections des CV, une étape de préparation et de prétraitement des données est cruciale. L'objectif est de normaliser l'ensemble des CV dans un format uniforme, facilitant ainsi les tâches d'annotation et d'entraînement du modèle de détection d'objets. Nous allons mettre en place un pipeline de prétraitement robuste utilisant des bibliothèques Python courantes en traitement d'images et en science des données.

4.2.2.1 Pipeline de prétraitement

Nous utiliserons principalement les bibliothèques suivantes :

— OpenCV (cv2) pour le traitement d'image

— NumPy pour les opérations sur les tableaux

— scikit-image pour des transformations d'image avancées

Voici les principales étapes de notre pipeline de prétraitement, avec des exemples de code pour chaque étape :

Étape 1 : Chargement et conversion des images
Concept : Cette étape consiste à charger une image à partir d'un fichier spécifié et à la convertir en niveaux de gris. L'objectif est de simplifier le traitement ultérieur en réduisant la complexité des données à une seule dimension de couleur (niveaux de gris), ce qui facilite l'analyse et l'extraction d'informations.

Étape 2 : Redimensionnement à une taille standardisée
Concept : Redimensionner les images à une taille standardisée est essentiel pour assurer une uniformité dans le traitement des données. Dans ce cas, les images sont ajustées à une taille typique de format A4 à 300 DPI, ce qui est couramment utilisé pour des documents imprimés. Cela garantit que toutes les images sont traitées de manière cohérente, ce qui est crucial pour la précision des modèles de détection d'objets.

Étape 3 : Correction de l'orientation
Concept : L'orientation des CV peut varier, et il est important de s'assurer que toutes les images sont alignées de manière cohérente. Cette étape utilise la détection de lignes dans l'image pour identifier l'orientation incorrecte et appliquer une rotation si nécessaire. Cela permet de corriger automatiquement les images mal orientées et assure une consistance dans la présentation des données pour le traitement ultérieur.

Étape 4 : Filtrage et suppression des bruits
Concept : Les images peuvent contenir du bruit, des artefacts ou des imperfections qui peuvent interférer avec le processus d'analyse. Pour améliorer la qualité des images, cette étape utilise des filtres pour réduire le bruit tout en préservant les détails importants. Cela permet d'améliorer la clarté des images et de rendre les caractéristiques d'intérêt plus discernables, ce qui est crucial pour la précision des modèles.

Étape 5 : Amélioration du contraste
Concept : L'amélioration du contraste est une technique qui ajuste la luminosité des pixels pour améliorer la distinction entre les objets et les détails dans une image. Cette étape utilise une égalisation d'histogramme adaptative pour ajuster la luminosité de manière adaptative, ce qui est particulièrement utile pour les images où les variations d'éclairage sont importantes. Cela permet de rendre les informations plus visibles et facilite l'identification des éléments dans les CV.

Étape 6 : Augmentation des données
Concept : L'augmentation des données est une pratique essentielle en apprentissage automatique pour enrichir l'ensemble de données d'entraînement. Cette étape génère plusieurs variantes de chaque image en appliquant des transformations telles que la rotation, la variation de luminosité et l'ajout de bruit gaussien. Cela permet de diversifier les données d'entraînement, améliorant ainsi la capacité du modèle à généraliser et à reconnaître une plus grande variété de situations et de conditions.

Pipeline complete :

Ce pipeline de prétraitement applique systématiquement une série de transformations à chaque CV, garantissant ainsi une uniformité dans le jeu de données. L'étape d'augmentation des données enrichit considérablement notre ensemble d'entraînement, améliorant potentiellement la robustesse et la généralisation de notre modèle de détection d'objets.

Il est important de noter que ce pipeline peut être ajusté en fonction des caractéristiques spécifiques de notre jeu de données de CV. Par exemple, si nous constatons que certains CV sont systématiquement surexposés, nous pourrions ajouter une étape de correction d'exposition avant l'amélioration du contraste.

Ces améliorations permettront d'optimiser davantage notre processus de prétraitement des CV, contribuant ainsi à l'efficacité globale de notre système de détection de zones textuelles dans les documents visuels.

Ces étapes de prétraitement sont essentielles pour créer un jeu de données de CV homogène et optimisé pour l'annotation dans CVAT, et pour le fine-tuning du modèle de détection d'objets Faster R-CNN. En appliquant ces techniques, nous améliorons la qualité des données d'entraînement, ce qui se traduit par de meilleures performances du modèle.

4.2.2.2 Annotation des données avec CVAT

CVAT (Computer Vision Annotation Tool) est un outil d'annotation d'images et de vidéos gratuit et open source basé sur le Web, utilisé pour étiqueter les données pour les algorithmes de vision par ordinateur permettant d'annoter manuellement des jeux de données d'images et de vidéos. Dans le cadre de cette étude de cas sur la détection des sections de CV, CVAT sera utilisé pour marquer les différentes zones d'intérêt dans les images de CV.

Voici les principales étapes de l'annotation des données avec CVAT :

Installation et Configuration de CVAT : Tout d'abord, il faut installer CVAT sur le système de travail. L'outil est disponible sous forme d'application web, ce qui facilite son déploiement et son utilisation. Une fois installé, CVAT doit être configuré avec les informations pertinentes pour le projet, comme les catégories d'annotation à utiliser.

Import des Images de CV : Les images de CV prétraitées seront importées dans CVAT. L'outil permet de charger des lots d'images en une seule fois, facilitant ainsi le processus d'annotation.

Définition des Catégories d'Annotation : Les différentes sections à détecter dans les CV (comme "Expérience", "Formation", "Compétences", etc.) seront définies en tant que catégories d'annotation dans CVAT. Chaque catégorie se verra attribuer un identifiant unique.

Annotation Manuelle des Sections : L'annotation proprement dite consiste à tracer des boîtes délimitant les différentes sections présentes dans chaque image de CV. CVAT offre une interface intuitive permettant de dessiner rapidement ces boîtes d'annotation et de les associer aux catégories correspondantes.

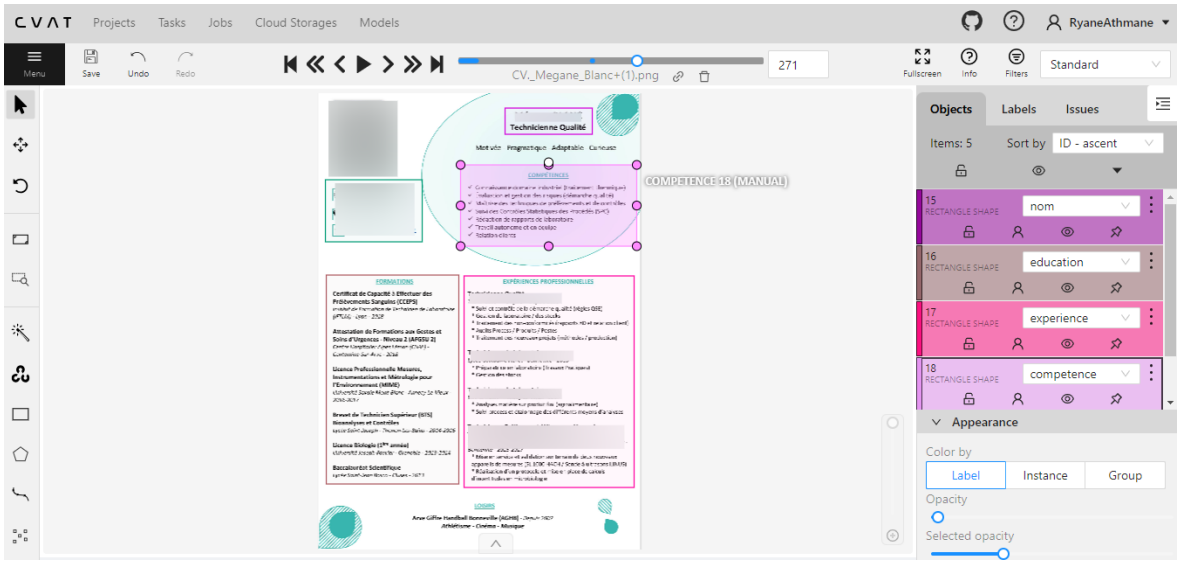


Figure 4.2 – Annotation des cv avec CVAT

Exportation des Annotations : Une fois l'annotation manuelle terminée, les données annotées pourront être exportées dans un format standard, comme le format JSON. Ce fichier d'annotations contiendra les informations sur les boîtes délimitant les sections, ainsi que leurs catégories respectives. Grâce à CVAT, l'équipe a pu annoter de manière efficace et cohérente un jeu de données représentatif de CV. Les catégories d'annotation définies, à savoir "Expérience", "Formation", "Compétences", "Profil", "Nom" et "Informations Personnelles", permettront d'entraîner un modèle de détection d'objets capable de localiser précisément ces différentes sections dans de nouveaux CV.

L'export du jeu de données annoté au format JSON facilitera l'étape suivante, à savoir le choix et le prétraitement des modèles de détection d'objets à évaluer sur cette tâche spécifique.

4.2.3 Configuration des modèles

4.2.3.1 Configuration initiale des architectures Faster R-CNN et Mask R-CNN

Pour obtenir des performances optimales avec les modèles Faster R-CNN et Mask R-CNN, il est crucial de configurer correctement ces architectures. Cette configuration initiale inclut le choix des hyperparamètres, les prétraitements des données, et l'ajustement des différents composants du modèle. Dans cette section, nous allons détailler les étapes clés de la configuration initiale des architectures Faster R-CNN et Mask R-CNN, ainsi que les choix des hyperparamètres qui influencent la qualité et la rapidité des résultats.

Configuration initiale des architectures Faster R-CNN et Mask R-CNN
La configuration initiale des architectures Faster R-CNN et Mask R-CNN est cruciale pour garantir que les modèles sont correctement préparés pour le fine-tuning et l'entraînement.

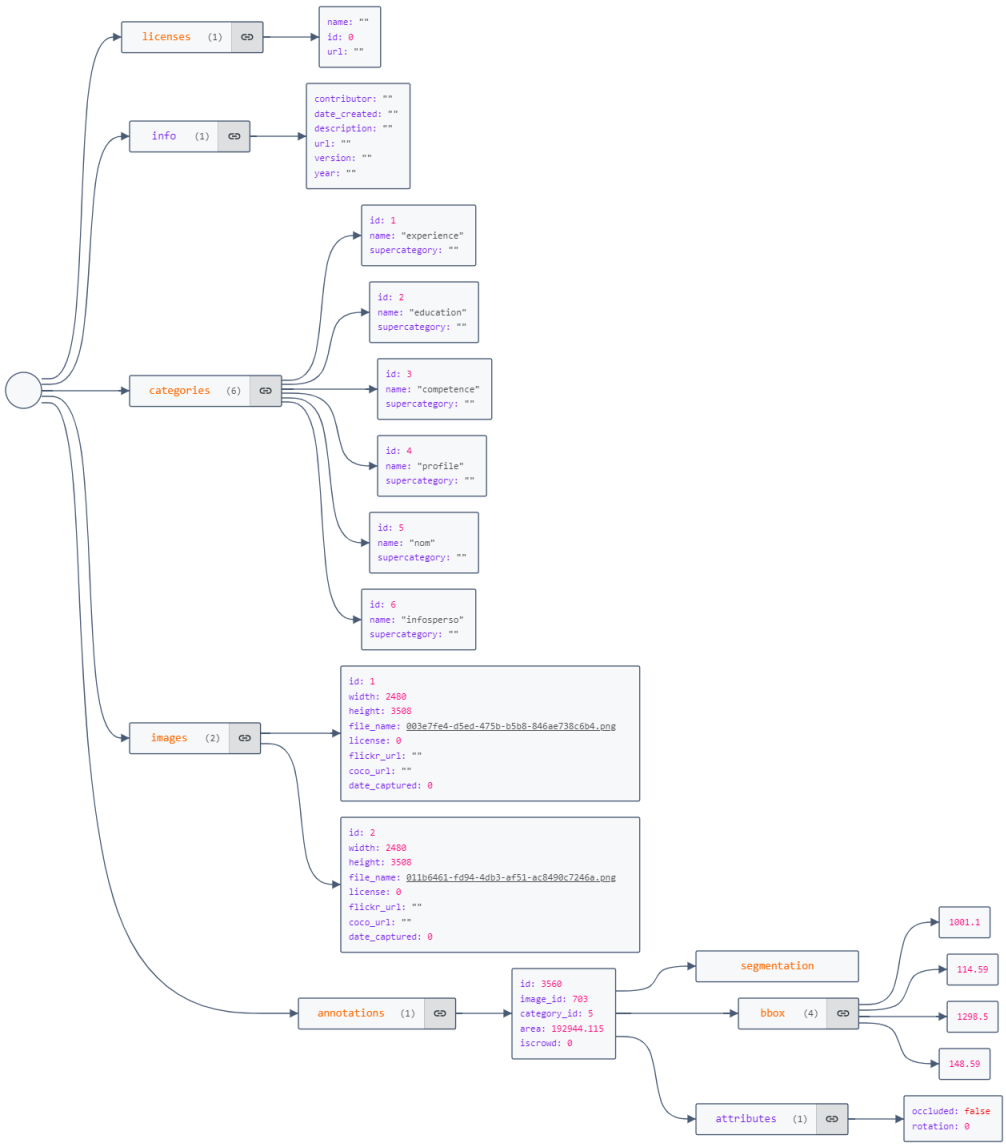


FIGURE 4.3 – Format des annotations json

Voici les étapes principales de cette configuration :

— **Sélection du Backbone (ou "Backbone Network")** Le backbone est le réseau de base utilisé pour extraire les caractéristiques (features) des images. Des réseaux populaires tels que ResNet, VGG, et Inception sont souvent utilisés en raison de leurs performances éprouvées sur des tâches de classification d'images.

La sélection du backbone dépend du compromis entre précision et rapidité : les modèles plus profonds sont ResNet offrent une meilleure précision mais au coût d'une vitesse d'inférence réduite.

— **Initialisation des Poids :** Les poids du backbone sont généralement initialisés avec des valeurs pré-entraînées sur des bases de données larges telles qu'ImageNet. Cela permet d'accélérer la convergence et d'améliorer les performances du modèle sur des

jeux de données spécifiques.

Les nouvelles couches ajoutées pour la détection (Region Proposal Network - RPN et les couches fully connected) sont initialisées avec des poids aléatoires ou des distributions spécifiques comme He ou Xavier initializations.

— **Configuration des Propositions de Régions (RPN) :** Le RPN génère des propositions de régions d'intérêt (Regions of Interest - RoI). Les hyperparamètres clés incluent le nombre d'anchor boxes, leurs tailles et ratios d'aspect.

Une configuration adaptée des anchor boxes est essentielle pour couvrir efficacement les différentes tailles et formes des sections du CV dans les images. Par exemple, des anchors avec des ratios de 1 :1, 2 :1 et 1 :2 sont utilisés.

— **RoI Pooling/RoI Align :** Pour Faster R-CNN, la couche RoI Pooling transforme les régions d'intérêt de tailles variables en cartes de caractéristiques de taille fixe. Mask R-CNN utilise RoI Align pour éviter la perte d'information due à la quantification.

La taille des sorties de ces couches doit être configurée (par exemple, 7x7 pour Faster R-CNN et 14x14 pour Mask R-CNN), assurant ainsi une compatibilité avec les couches fully connected suivantes.

4.2.3.2 Choix des hyperparamètres et des prétraitements

Le choix des hyperparamètres influence directement la performance et l'efficacité des modèles Faster R-CNN et Mask R-CNN :

— **Taux d'apprentissage (Learning Rate) :** Un taux d'apprentissage approprié est crucial pour assurer une convergence stable et rapide. Un planning de décroissance du taux d'apprentissage (par exemple, réduction du taux de moitié toutes les quelques époques) est souvent utilisé. Un taux d'apprentissage typique peut commencer à 0,001 et être réduit progressivement.

— **Taille des batchs (Batch Size) :** La taille des batchs influence à la fois la stabilité de l'entraînement et la mémoire nécessaire. Des batchs plus grands permettent une estimation plus stable des gradients mais nécessitent plus de mémoire GPU. Une taille de batch courante pour ces modèles se situe entre 1 et 16, en fonction de la capacité du GPU.

— **Nombre d'itérations/époques (Iterations/Epochs) :** Le nombre d'itérations ou d'époques détermine combien de fois le modèle verra chaque exemple de données. Un nombre trop faible peut mener à un sous-ajustement, tandis qu'un nombre trop élevé peut entraîner un sur-ajustement. Typiquement, on peut commencer avec 10 à 20 époques pour un ajustement initial.

4.2.3.3 Étapes du fine-tuning

Le fine-tuning des modèles Faster R-CNN et Mask R-CNN implique plusieurs étapes clés, allant de l'entraînement initial sur un dataset personnalisé à l'évaluation des performances

sur un dataset de validation. Dans cette section, nous décrivons de manière technique détaillée de chacune de ces étapes.

Préparation des Données :

- **Importation des Données Annotées :** Les données annotées au format JSON (issues de CVAT) sont importées et converties dans un format compatible avec les frameworks d'entraînement (notre cas, COCO format car nous utilisons PyTorch).
- **Chargement des Données :** Utilisation de DataLoader pour charger les données en batchs, assurant une lecture efficace depuis le disque et une gestion adéquate de la mémoire.
- **Augmentation des Données :** Application des techniques d'augmentation des données pendant le chargement pour augmenter la robustesse du modèle. Cela inclut la rotation aléatoire, le recadrage, le retournement horizontal et les ajustements de luminosité et de contraste.

Configuration du Modèle :

- **Backbone :** Sélection et chargement du backbone pré-entraîné (ResNet-50) avec des poids initialisés sur ImageNet.
- **RPN (Region Proposal Network) :** Configuration des anchor boxes avec différentes échelles et ratios pour capturer une variété de tailles et formes d'objets dans les CV.
- **RoI Pooling/RoI Align :** Mise en place de RoI Align pour Mask R-CNN afin de préserver la précision des régions d'intérêt.
- **Masques :** Pour Mask R-CNN, ajout de la branche de segmentation des masques, avec une résolution appropriée.

Phase d'Entraînement Initial :

- **Taux d'Apprentissage :** Initialisation du taux d'apprentissage, avec un plan de décroissance pour réduire progressivement le taux d'apprentissage.
- **Optimiseur :** Utilisation de l'optimiseur Adam ou SGD, avec un réglage initial des paramètres tels que le momentum et la régularisation L2 (poids de décroissance).
- **Entraînement en Deux Phases : Phase 1 :** Entraînement des couches nouvellement ajoutées (RPN, RoI heads) avec le backbone fixé (non entraîné).
  Phase 2 : Fine-tuning de l'ensemble du réseau, incluant le backbone, avec un taux d'apprentissage réduit.

Ajustement des hyperparamètres et optimisation

- **Grid Search et Random Search :** Mise en œuvre de techniques de recherche d'hyperparamètres pour trouver les valeurs optimales de taux d'apprentissage, taille des batchs, et nombre d'ancres.
- **Validation Croisée :** Utilisation de la validation croisée pour évaluer la performance des différentes combinaisons d'hyperparamètres sur des sous-ensembles du dataset d'entraînement.

Optimisation :

- **Scheduling du Taux d'Apprentissage :** Implémentation de techniques de scheduling telles que ReduceLROnPlateau ou Cosine Annealing pour ajuster dynamiquement le taux d'apprentissage pendant l'entraînement.

— **Early Stopping :** Utilisation de l'early stopping pour arrêter l'entraînement lorsque la performance sur le dataset de validation ne s'améliore plus après un certain nombre d'époques.

— **Gradient Clipping :** Application du gradient clipping pour éviter les problèmes de gradient explosif, surtout dans le cas des architectures profondes.

Métriques d'Évaluation :

— **mAP (mean Average Precision) :** Calcul du mAP pour évaluer la précision des détections et des segments. Le mAP est souvent calculé à différents seuils d'IoU (Intersection over Union), tels que 0.75 (mAP@75) et 0.90 (mAP@90).

— **AP par Classe :** Analyse de l'Average Precision (AP) par classe pour identifier les classes où le modèle performe bien et celles nécessitant des améliorations.

Affinement du Modèle :

— **Analyse des Faux Positifs et Faux Négatifs :** Étude des faux positifs et faux négatifs pour comprendre les types d'erreurs commises par le modèle et ajuster les stratégies d'augmentation des données et les hyperparamètres en conséquence.

— **Ré-annotation et Augmentation des Données :** Si nécessaire, ré-annotation de certaines données mal interprétées et augmentation des données pour couvrir les cas d'erreurs fréquents.

En suivant ces étapes, nous pouvons affiner et optimiser les modèles Faster R-CNN et Mask R-CNN pour obtenir des performances élevées sur des tâches de détection et de segmentation d'objets spécifiques, telles que la reconnaissance des sections des CV.

4.2.4 Résultats et Comparaison

Pour évaluer les performances des modèles Faster R-CNN et Mask R-CNN, nous avons mené des expériences sur un GPU T4 sur Google Colab. Les métriques principales, notamment la précision et le temps d'inférence, ont été prises en compte. Les résultats obtenus présentent des nuances significatives.

Paramètres des modèles

Paramètre	Faster R-CNN	Mask R-CNN
Nombre d'époques d'entraînement	10	10
Taux d'apprentissage initial	0.001	0.001
Taux d'apprentissage pour le fine-tuning	0.0001	0.0001
Taille du batch	4	4
Taille de l'image d'entrée	2480x3508 pixels	2480x3508 pixels
Backbones	ResNet-50	ResNet-50

Table 4.1 – Comparaison des paramètres de Faster R-CNN et Mask R-CNN

Paramètre	Faster R-CNN	Mask R-CNN
Précision	87.5%	81.2%
Rappel	85.3%	72.0%
mAP@0.75	86.1%	81.4%
mAP@0.90	75.4%	69.3%
Temps d'inférence par image	0.06s	0.14s

TABLE 4.2 – Comparaison des performances de Faster R-CNN et Mask R-CNN

4.2.4.1 Analyse des résultats des deux modèles

4.2.4.2 Choix du modèle optimal en fonction des résultats

L'évaluation des performances des modèles Faster R-CNN et Mask R-CNN nous permet de sélectionner le modèle le plus adapté à notre tâche spécifique de reconnaissance des sections de CV. Cette décision repose sur une analyse comparative des différentes métriques de performance obtenues lors des expériences et sur les exigences pratiques de notre application.

Critères de Sélection :

- **Précision et Rappel :**
  Faster R-CNN montre une meilleure précision (87.5%) et rappel (85.3%) comparé à Mask R-CNN (81.2% de précision et 72.0% de rappel). Cela suggère que Faster R-CNN est plus efficace pour identifier correctement les sections de CV sans les manquer ou générer trop de fausses alertes.
- **mAP (Mean Average Precision) :**
  À un seuil IoU de 0.75, Faster R-CNN atteint un mAP de 86.1%, ce qui est supérieur au 81.4% de Mask R-CNN. De même, à un seuil IoU plus strict de 0.90, Faster R-CNN (75.4%) surpasse encore Mask R-CNN (69.3%). Cela indique que Faster R-CNN a une meilleure performance en termes de précision globale et de robustesse à des exigences de correspondance plus strictes.
- **Temps d'Inférence :**
  Faster R-CNN est plus rapide avec un temps d'inférence de 0.06 secondes par image, contre 0.14 secondes pour Mask R-CNN. Cette différence de vitesse est cruciale dans des scénarios où le temps de traitement est un facteur limitant, comme dans des applications en temps réel ou à grande échelle.
- **Complexité des Tâches :**
  Mask R-CNN ajoute une dimension supplémentaire avec la segmentation des masques, ce qui est utile lorsque la localisation précise des pixels est essentielle. Cependant, pour notre tâche de reconnaissance des sections de CV, qui semble être bien adressée par une détection de boîte englobante fournie par Faster R-CNN, cette capacité supplémentaire n'est peut-être pas nécessaire.

Compte tenu des résultats et des critères ci-dessus, **Faster R-CNN semble être le modèle optimal** pour notre application spécifique :

- Il offre une meilleure précision et un meilleur rappel, ce qui est crucial pour minimiser

les erreurs de détection.
- Son temps d'inférence plus court permet une utilisation plus efficace dans des contextes où la rapidité est essentielle.
- Bien que Mask R-CNN offre des capacités de segmentation des masques, pour notre tâche de détection de sections de CV, les bénéfices supplémentaires de Mask R-CNN ne compensent pas ses performances inférieures en termes de précision et de temps de traitement.

En conclusion, Faster R-CNN est recommandé pour sa performance globale supérieure et sa rapidité, ce qui le rend plus adapté à notre objectif de reconnaissance des sections de CV.

4.2.5 Comparaison de plusieurs OCR

4.2.5.1 Présentation des différents outils OCR : Tesseract, EasyOCR, PaddleOCR

Tesseract OCR

Tesseract est l'un des outils OCR les plus connus et les plus utilisés dans le monde de l'extraction de texte. Développé par HP et maintenant maintenu par Google, Tesseract est un logiciel open-source qui prend en charge plus de 100 langues. Il est particulièrement apprécié pour sa robustesse et sa capacité à gérer des documents complexes avec des mises en page variées.

- **Langues et Scripts Supportés :** Tesseract peut reconnaître une large gamme de langues et de scripts, ce qui le rend très versatile pour les projets internationaux.
- **Modes de Fonctionnement :** Il prend en charge différents modes de reconnaissance, y compris la reconnaissance de texte brut et la reconnaissance de texte structuré avec des tables et des colonnes.
- **Facilité d'Intégration :** Tesseract est facilement intégrable dans diverses applications grâce à ses bindings disponibles pour plusieurs langages de programmation comme Python et C++.

EasyOCR

EasyOCR est un outil OCR plus récent mais qui gagne en popularité grâce à sa simplicité et ses performances. Développé par Jaided AI, cet outil est également open-source et se distingue par sa capacité à reconnaître le texte en utilisant des modèles de deep learning, ce qui lui permet de traiter des documents et des images avec une précision élevée.

- **Technologie de Reconnaissance :** Utilise des réseaux neuronaux convolutionnels (CNN) pour extraire les caractères des images, offrant ainsi une bonne performance même sur des textes complexes et manuscrits.
- **Langues Supportées :** EasyOCR prend en charge plus de 80 langues, y compris les scripts non-latins comme le chinois, le japonais et le coréen.
- **Simplicité d'Utilisation :** Conçu pour être facile à utiliser, avec une interface Python simple et des dépendances minimales, ce qui le rend accessible même pour les utilisateurs non experts.

PaddleOCR

PaddleOCR fait partie du projet PaddlePaddle développé par Baidu, un framework open-source de deep learning. PaddleOCR est conçu pour offrir des performances OCR élevées, en particulier pour les langues asiatiques. Il combine plusieurs modèles de deep learning pour fournir des capacités de reconnaissance de texte puissantes et flexibles.

- **Architecture Avancée :**Utilise une architecture de deep learning sophistiquée qui combine plusieurs modules, y compris des modèles pour la détection de texte, la reconnaissance de caractères et la post-traitement des résultats.
- **Support Multilingue :** Optimisé pour une large gamme de langues, avec un accent particulier sur les langues asiatiques complexes.
- **Performances Optimisées :** Conçu pour tirer parti des GPU pour accélérer le traitement et offrir une reconnaissance de texte très rapide et précise.

4.2.5.2 Critères de comparaison : précision, vitesse, facilité d'utilisation

Pour choisir l'outil OCR le plus adapté à notre projet de traitement de CV, il est essentiel d'évaluer chaque option selon plusieurs critères clés :

- **Précision**
  La précision de l'OCR est cruciale pour assurer que le texte extrait correspond exactement à ce qui est présent dans l'image. Une haute précision réduit la nécessité de corrections manuelles après l'extraction.
  - **Tesseract** est réputé pour sa précision sur des textes imprimés de bonne qualité. Cependant, il peut rencontrer des difficultés avec des documents de mauvaise qualité ou des mises en page complexes.
  - **EasyOCR** offre une bonne précision, surtout sur les textes manuscrits ou les polices non conventionnelles, grâce à son utilisation de modèles de deep learning.
  - **PaddleOCR** est particulièrement performant sur les textes complexes et les langues asiatiques, bénéficiant de ses modèles sophistiqués de deep learning.
- **Vitesse**
  La vitesse de traitement est un facteur important, surtout pour les applications en temps réel ou le traitement de grands volumes de documents.
  - **Tesseract** peut être relativement lent, surtout lorsqu'il doit traiter des documents volumineux ou des mises en page complexes.
  - **EasyOCR** est conçu pour être rapide, en exploitant les capacités des GPU pour accélérer le traitement.
  - **PaddleOCR** est optimisé pour des performances rapides, surtout lorsqu'il est exécuté sur du matériel compatible avec le deep learning, comme les GPU.
- **Facilité d'Utilisation**
  L'accessibilité de l'outil, y compris la simplicité de son intégration et de son utili-

sation, est essentielle pour garantir une adoption facile par les développeurs et les utilisateurs finaux.

- **Tesseract** est bien documenté et dispose de nombreux bindings pour divers langages de programmation, mais il peut nécessiter des ajustements et une configuration pour des tâches spécifiques.
- **EasyOCR** est très facile à utiliser avec une API Python intuitive, ce qui le rend idéal pour les développeurs à la recherche d'une solution rapide et efficace.
- **PaddleOCR** offre une bonne flexibilité grâce à ses capacités avancées et à sa compatibilité avec les projets PaddlePaddle, mais peut nécessiter une courbe d'apprentissage pour les utilisateurs non familiers avec le framework.

4.2.6 Synthèse et Choix de l'Outil OCR

En résumé, le choix de l'outil OCR dépend des besoins spécifiques de notre projet de traitement de CV. Voici une synthèse de l'évaluation :

- **Précision :** Si la précision est la priorité, **EasyOCR** et **PaddleOCR** sont préférables pour leur robustesse sur des textes variés, y compris les manuscrits et les scripts non-latins.
- **Vitesse :** Pour des applications nécessitant une rapidité d'exécution, **EasyOCR** et **PaddleOCR** offrent de meilleures performances, surtout lorsqu'ils sont utilisés avec du matériel GPU.
- **Facilité d'Utilisation : EasyOCR** se distingue par sa simplicité d'utilisation et son API conviviale, ce qui le rend attrayant pour les développeurs cherchant à intégrer rapidement une solution OCR.

Dans le cadre de notre projet de reconnaissance de sections de CV, où la précision de l'extraction des informations est critique, **EasyOCR** est recommandé en raison de son équilibre entre précision, vitesse et facilité d'utilisation. PaddleOCR pourrait également être considéré si nous devons traiter des documents dans des langues asiatiques ou avec des scripts complexes.

4.3 Conclusion

Ce chapitre met en lumière les choix stratégiques cruciaux pour l'automatisation de la détection d'objets dans les CV, en privilégiant les modèles Faster R-CNN et Mask R-CNN pour leur capacité à offrir une détection précise des sections pertinentes et une segmentation détaillée des objets. L'intégration de ces techniques avancées avec des outils OCR comme Tesseract, EasyOCR et PaddleOCR permet non seulement une extraction efficace du texte mais aussi une gestion optimisée des données dans les systèmes de ressources humaines. Cette approche promet de réduire les efforts manuels tout en améliorant la fiabilité et la vitesse des processus de gestion de candidatures.

Dans le prochain chapitre, nous mettrons en pratique les concepts et techniques abordés en appliquant des modèles de NLP (Natural Language Processing) aux données textuelles extraites par l'OCR. Nous démontrerons comment les modèles NLP peuvent être utilisés pour analyser et interpréter ces données textuelles, permettant des applications telles que l'extraction d'informations. Cette transition de la vision par ordinateur et de l'OCR vers le NLP illustrera comment les technologies complémentaires peuvent être intégrées pour créer des solutions plus complètes et efficaces pour le traitement et l'analyse des données textuelles.

Chapitre 5

Optimisation par Fine-Tuning des Modèles de Traitement automatique du langage naturel (NLP) pour la reconnaissance d'entité nommée (NER) : Approches et Contributions

5.1 Introduction

Ce chapitre explore l'application des techniques avancées de traitement automatique du langage naturel (NLP) pour la reconnaissance d'entités nommées (NER) dans les CV. La reconnaissance d'entités nommées est une tâche clé dans le domaine du NLP, consistant à identifier et classifier les éléments textuels en catégories prédéfinies telles que les noms de personnes, les organisations, les dates, et les lieux. Dans le contexte de l'analyse de CV.

Nous aborderons deux aspects principaux dans ce chapitre : La préparation et l'annotation des données et le fine-tuning des modèles de NLP. L'objectif est de démontrer comment ces approches peuvent être combinées pour créer un système robuste et performant.

5.2 Fine-Tuning pour la Tâche NER

5.2.1 Préparation des Données

La première étape pour fine-tuner un modèle de NLP pour la tâche de reconnaissance d'entités nommées (NER) consiste à préparer un corpus de données annotées. Pour notre

projet de traitement de CV, Nous avons utiliser les meme données utiliser pour le fine tuning du modèle de computer vision ou grâce au modèle Faster R-CNN et de l'OCR EasyOCR nous avons pu récupère les données textuelle de chaque section de chaque CV de notre dataset.

5.2.2 Annotation des Textes pour la Tâche NER

L'annotation est réalisée pour identifier les entités nommées pertinentes dans les CV, telles que les noms de personnes, les adresses, les dates, les titres de postes, les noms d'entreprises, et les compétences. Nous utilisons des outils d'annotation comme BRAT ou Prodigy pour marquer les entités dans le texte. Chaque entité est étiquetée avec une catégorie spécifique (par exemple, "Nom", "Entreprise", "Compétence").

Pour l'outil utulisee nous avons opte pour Label Studio un outil open-source qui prend en charge une large gamme de tâches d'annotation, y compris NER. Il offre une interface intuitive et de nombreuses options de configuration.

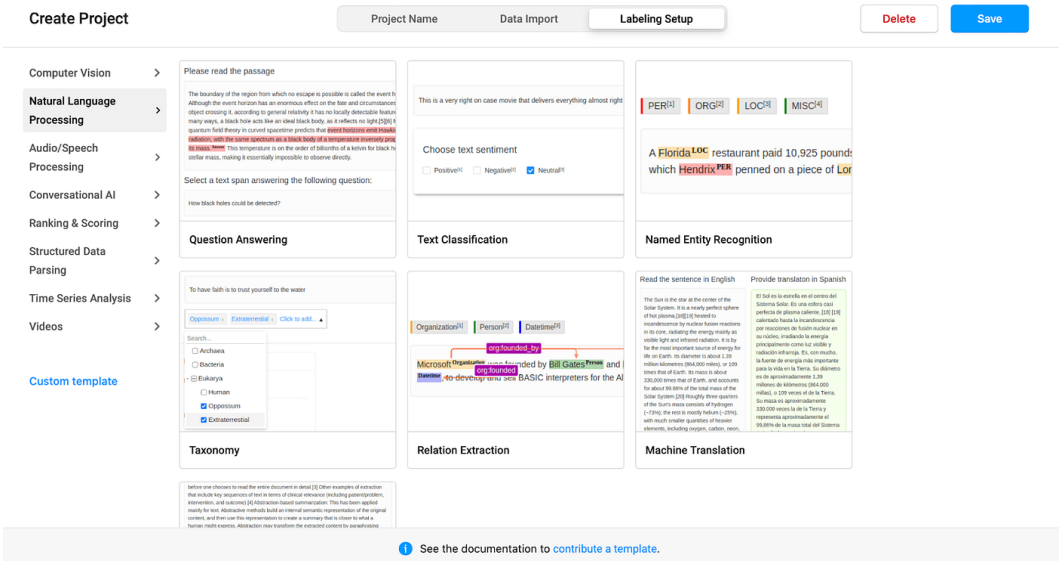


Figure 5.1 – Outil d'annotation Label-Studio

5.2.2.1 Format d'annotation

Le format BIO est un schéma de codage populaire et efficace pour l'annotation des entités nommées dans les tâches de reconnaissance d'entités nommées (NER). Il offre une méthode claire et structurée pour marquer les entités, facilitant ainsi l'entraînement et l'évaluation des modèles de NLP. En combinant le format BIO avec des outils d'annotation automatisés et des processus de vérification rigoureux, il est possible de créer des datasets de haute qualité pour le fine-tuning des modèles de NER.

- Expérience :
  - Poste
  - Période (Date de début, Date de fin)
  - Entreprise
  - Tâches
- Éducation :
  - Établissement
  - Période
  - Formation
- Compétences :
  - Liste de compétences directement
- Informations personnelles :
  - Nom
  - Prénom
  - Adresse email
  - Adresse postale
  - Numéro de téléphone

Une fois l'annotation terminée, un processus de vérification est effectué pour garantir la qualité et la cohérence des annotations. Cela peut impliquer plusieurs cycles de révision pour minimiser les erreurs. Le processus inclut une double Annotation ou chaque document est re vérifier pour assurer la cohérence et réduire les erreurs.

Voici un exemple de CV annoté avec des balises NER en utilisant le format BIO (Beginning, Inside, Outside).

Explications des annotations

- **Nom :** B-Nom pour le début du nom, I-Nom pour l'intérieur du nom.
- **Email :** B-Email pour l'email.
- **Adresse :** B-Adresse pour le début de l'adresse, `I-Adresse` pour l'intérieur de l'adresse.
- **Téléphone :** B-Telephone pour le début du numéro de téléphone, I-Telephone pour l'intérieur du numéro de téléphone.
- **Expérience :**
  - B-Experience-Poste pour le début du poste, I-Experience-Poste pour l'intérieur du poste.
  - B-Experience-Entreprise pour le début de l'entreprise, I-Experience-Entreprise pour l'intérieur de l'entreprise.
  - B-Experience-Periode pour le début de la période, I-Experience-Periode pour l'intérieur de la période.

```
Nom: John Doe
Adresse email: john.doe@example.com
Adresse postale: 123 Rue de Paris, 75001 Paris
Numéro de téléphone: +33 6 12 34 56 78
Lien LinkedIn: linkedin.com/in/johndoe

Profil:
Développeur logiciel avec 5 ans d'expérience dans le développement d'applications web et mobiles. Passionné

Expérience:
- Développeur Frontend chez TechCorp
  Période: Janvier 2019 - Présent
  Tâches: Conception et développement de l'interface utilisateur pour des applications web.

- Ingénieur Logiciel chez WebSolutions
  Période: Juin 2015 - Décembre 2018
  Tâches: Développement d'applications web, maintenance de systèmes existants.

Éducation:
- Master en Informatique à l'Université de Paris
  Période: Septembre 2013 - Juin 2015

Compétences:
- Langages de programmation: JavaScript, Python, Java
- Frameworks: React, Angular, Django
- Bases de données: MySQL, MongoDB
```

Figure 5.2 – Données textuelle de CV

  - B-Experience-Taches pour le début des tâches, I-Experience-Taches pour l'intérieur des tâches.
- **Éducation :**
  - B-Education-Formation pour le début de la formation, I-Education-Formation pour l'intérieur de la formation.
  - B-Education-Etablissement pour le début de l'établissement, I-Education-Etablissement pour l'intérieur de l'établissement.
  - B-Education-Periode pour le début de la période, I-Education-Periode pour l'intérieur de la période.
- **Compétences :** B-Competence pour chaque compétence.

En suivant ces étapes, nous avons créé un corpus de données annotées de qualité, prêt à être utilisé pour le fine-tuning des modèles de NLP pour la tâche de NER.

5.3 Fine-Tuning des Modèles

5.3.1 Fine-Tuning de CamemBERT

Le fine-tuning de CamemBERT pour la tâche de reconnaissance d'entités nommées (NER) consiste à adapter le modèle préentraîné pour qu'il performe sur des données spécifiques annotées en format BIO (Beginning, Inside, Outside). Cette méthode permet au modèle de reconnaître et de classer les entités nommées dans un texte, telles que les noms de personnes, de lieux, d'organisations, etc. Voici les étapes détaillées du processus :

1. **Préparation des Données :**

```
John B-Nom
Doe I-Nom
john.doe@example.com B-Email
123 B-Adresse
Rue I-Adresse
de I-Adresse
Paris I-Adresse
, I-Adresse
75001 I-Adresse
Paris I-Adresse
+33 B-Telephone
6 I-Telephone
12 I-Telephone
34 I-Telephone
56 I-Telephone
78 I-Telephone
linkedin.com/in/johndoe B-Lien
Développeur B-Experience-Poste
Frontend I-Experience-Poste
chez B-Experience-Entreprise
TechCorp I-Experience-Entreprise
Janvier B-Experience-Periode
2019 I-Experience-Periode
- I-Experience-Periode
Présent I-Experience-Periode
Conception B-Experience-Taches
et I-Experience-Taches
développement I-Experience-Taches
de I-Experience-Taches
```

FIGURE 5.3 – Anntation format BIO

- **Vérification du Format BIO :**
  - Les données doivent être annotées suivant le format BIO, où chaque mot est étiqueté comme étant le début (B) ou l'intérieur (I) d'une entité, ou en dehors (O) de toute entité. Ce format permet une granularité fine dans l'identification des entités.
  - Assurez-vous que chaque phrase est correctement segmentée et alignée avec ses annotations pour éviter les erreurs d'étiquetage.
- **Division du Corpus :**
  - Il est essentiel de diviser le corpus en ensembles d'entraînement, de validation et de test. Généralement, on utilise environ 70% des données pour l'entraînement, 15% pour la validation et 15% pour le test. Cette division permet de mesurer la performance du modèle de manière fiable et de détecter les problèmes de surapprentissage (overfitting).

2. **Chargement du Modèle Préentraîné :**

- **Utilisation de la Bibliothèque transformers de Hugging Face :**
  - La bibliothèque transformers facilite l'utilisation des modèles préentraînés tels que CamemBERT. Ces modèles sont déjà entraînés sur de vastes corpus de données et peuvent être facilement adaptés à des tâches spécifiques comme la NER.

- **Importation de CamembertForTokenClassification et CamembertTokenizer :**
  - CamembertForTokenClassification est une variante de CamemBERT adaptée pour les tâches de classification de tokens. CamembertTokenizer est utilisé pour transformer les textes en tokens compatibles avec le modèle CamemBERT.

3. **Préparation des Données pour le Modèle :**

   - **Tokenisation des Textes :**
     - La tokenisation est le processus de découpage du texte en unités compréhensibles par le modèle. Utiliser CamembertTokenizer assure une compatibilité totale avec CamemBERT.
     - Il est crucial d'aligner correctement les tokens avec les annotations BIO, car une mauvaise alignement peut entraîner des erreurs de classification.
   - **Création des DataLoaders :**
     - Les DataLoaders, fournis par `torch.utils.data.DataLoader`, permettent de charger les données en lots pendant l'entraînement, ce qui est essentiel pour la gestion de la mémoire et l'efficacité de l'entraînement. Ils facilitent également la permutation aléatoire des données et le batching.

4. **Définition de la Fonction de Perte et de l'Optimiseur :**

   - **Utilisation de AdamW :**
     - AdamW est une variante de l'optimiseur Adam qui inclut une correction de la régularisation L2 (décroissance de poids). Cet optimiseur est bien adapté aux modèles de réseaux de neurones profonds et permet une convergence rapide et stable.
   - **Définition de la CrossEntropyLoss :**
     - CrossEntropyLoss est couramment utilisée pour les tâches de classification. Elle mesure la différence entre la distribution prédite par le modèle et la distribution réelle des classes, ce qui est crucial pour ajuster les poids du modèle de manière appropriée.

5. **Entraînement du Modèle :**

   - **Configuration des Paramètres d'Entraînement :**
     - Les paramètres tels que le nombre d'époques, le taux d'apprentissage et la gestion du scheduler de taux d'apprentissage doivent être soigneusement configurés. Par exemple, le scheduler peut réduire le taux d'apprentissage lorsque la performance sur l'ensemble de validation stagne, aidant à éviter le surapprentissage.
   - **Boucles d'Entraînement :**
     - Pendant chaque époque, le modèle passe par toutes les données d'entraînement. Chaque passe avant (forward pass) calcule les prédictions du modèle, et la passe arrière (backward pass) ajuste les poids en fonction de la perte calculée.

- Une évaluation régulière sur l'ensemble de validation est essentielle pour suivre la performance du modèle et ajuster les hyperparamètres si nécessaire.

6. **Évaluation et Test :**

   - **Évaluation sur l'Ensemble de Test :**
     - Après l'entraînement, évaluer le modèle sur un ensemble de test non vu permet de mesurer sa généralisation. Les métriques comme la précision, le rappel et le score F1 sont calculées pour chaque catégorie d'entités, offrant une vue détaillée des performances du modèle.
   - **Ajustement des Hyperparamètres :**
     - En fonction des résultats de l'évaluation, il peut être nécessaire d'ajuster les hyperparamètres tels que le taux d'apprentissage, le nombre d'époques ou la taille des lots (batch size) pour optimiser les performances du modèle.

7. **Sauvegarde du Modèle :**

   - **Sauvegarde du Modèle Entraîné et du Tokenizer :**
     - Une fois le modèle entraîné, il est important de le sauvegarder ainsi que le tokenizer utilisé. Cela permet une utilisation future sans nécessiter de réentraînement.
   - **Exportation des Résultats et Prédictions :**
     - Les résultats et les prédictions doivent être exportés pour analyse et comparaison. Cela inclut les métriques de performance et les prédictions pour chaque entité nommée, offrant une base pour une analyse plus approfondie et des comparaisons avec d'autres modèles ou méthodes.

5.3.2 Fine-Tuning d'un Modèle LLM

Le fine-tuning de Mistral 7B pour la tâche de reconnaissance d'entités nommées (NER) implique l'adaptation d'un modèle de langage de grande envergure à des données spécifiques annotées. Cette procédure permet au modèle d'extraire et de classer des entités nommées dans un texte, telles que les noms de personnes, de lieux, d'organisations, etc. Voici les étapes détaillées du processus :

1. **Préparation des Données :**

   - **Assurer que les données annotées en format BIO sont correctement structurées :**
     - Les données doivent suivre le format BIO (Beginning, Inside, Outside), où chaque mot est étiqueté comme étant le début (B) ou l'intérieur (I) d'une entité, ou en dehors (O) de toute entité. Ce format offre une granularité fine pour l'identification des entités.

- Une vérification minutieuse est nécessaire pour s'assurer que chaque annotation est correctement alignée avec les tokens correspondants dans le texte.

- **Division des données en ensembles d'entraînement, de validation et de test :**
  - Diviser les données est crucial pour évaluer de manière fiable les performances du modèle. En général, 70% des données sont utilisées pour l'entraînement, 15% pour la validation et 15% pour le test.
  - Cette division permet de mesurer la capacité du modèle à généraliser à de nouvelles données et d'éviter le surapprentissage (overfitting).

2. **Chargement du Modèle Préentraîné :**

   - **Utilisation de la bibliothèque transformers de Hugging Face :**
     - La bibliothèque `transformers` facilite l'utilisation et l'adaptation des modèles de langage préentraînés tels que Mistral 7B. Ces modèles sont déjà entraînés sur de vastes corpus de données et peuvent être adaptés à des tâches spécifiques comme la NER.
   - **Importation de `AutoModelForCausalLM` et `AutoTokenizer` :**
     - `AutoModelForCausalLM` est utilisé pour les tâches de modélisation du langage causal, et `AutoTokenizer` pour la tokenisation des textes. Leur utilisation conjointe permet de transformer les textes en tokens compatibles avec le modèle Mistral 7B.
     - La figure 5.4 montre le processus de chargement du modèle Mistral 7B.

```python
from transformers import AutoModelForCausalLM, AutoTokenizer
model_name = "mistralai/Mistral-7B-v0.1"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)
```

Figure 5.4 – Chargement du Modèle Mistral 7B

3. **Préparation des Données pour le Modèle :**

   - **Tokenisation des Textes :**
     - La tokenisation est le processus de découpage du texte en unités compréhensibles par le modèle. Utiliser `AutoTokenizer` assure une compatibilité totale avec Mistral 7B.
     - Il est crucial d'aligner correctement les tokens avec les annotations BIO pour éviter les erreurs de classification.
   - **Création des DataLoaders :**
     - Les `DataLoaders`, fournis par `torch.utils.data.DataLoader`, permettent de charger les données en lots pendant l'entraînement, ce qui est essentiel pour la gestion de la mémoire et l'efficacité de l'entraînement.

— Ils facilitent également la permutation aléatoire des données et le batching, améliorant ainsi la robustesse du modèle.

4. **Définition de la Fonction de Perte et de l'Optimiseur :**

   — **Utilisation de `AdamW` :**
      — `AdamW` est une variante de l'optimiseur Adam avec une correction de la régularisation L2 (décroissance de poids). Cet optimiseur est bien adapté aux modèles de réseaux de neurones profonds, permettant une convergence rapide et stable.
   — **Définition de la `CrossEntropyLoss` :**
      — La `CrossEntropyLoss` est couramment utilisée pour les tâches de classification. Elle mesure la différence entre la distribution prédite par le modèle et la distribution réelle des classes, ajustant ainsi les poids du modèle de manière appropriée.

5. **Entraînement du Modèle :**

   — **Configuration des Paramètres d'Entraînement :**
      — Les paramètres tels que le nombre d'époques, le taux d'apprentissage et la gestion du scheduler de taux d'apprentissage doivent être soigneusement configurés. Par exemple, un scheduler peut réduire le taux d'apprentissage lorsque la performance sur l'ensemble de validation stagne, aidant à éviter le surapprentissage.
   — **Boucles d'Entraînement :**
      — Pendant chaque époque, le modèle passe par toutes les données d'entraînement. Chaque passe avant (forward pass) calcule les prédictions du modèle, et la passe arrière (backward pass) ajuste les poids en fonction de la perte calculée.
      — L'utilisation d'outils comme `Trainer` de Hugging Face simplifie grandement le processus d'entraînement. La figure 5.5 montre un exemple de script de fine-tuning.
   — **Évaluation Régulière :**
      — Évaluer régulièrement la performance sur l'ensemble de validation permet de détecter et d'éviter le surapprentissage. Cela aide également à ajuster les hyperparamètres en fonction des performances observées.
   — **Optimisation par Quantification :**
      — La quantification transforme le modèle en un format plus léger tout en conservant une performance élevée, ce qui est particulièrement utile pour les déploiements sur des machines avec moins de ressources. Cela améliore l'efficacité et réduit l'utilisation de la mémoire.

6. **Évaluation et Test :**

   — **Évaluation sur l'Ensemble de Test :**

```python
from transformers import Trainer, TrainingArguments
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=4,
    num_train_epochs=3,
    weight_decay=0.01,
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=your_train_dataset,
    eval_dataset=your_eval_dataset,
)
trainer.train()
```

Figure 5.5 – Exécution du Fine-Tuning du Mistral 7B

- Après l'entraînement, évaluer le modèle sur un ensemble de test non vu permet de mesurer sa capacité de généralisation. Les métriques comme la précision, le rappel et le score F1 sont calculées pour chaque catégorie d'entités, offrant une vue détaillée des performances du modèle.

- **Ajustement des Hyperparamètres :**
  - En fonction des résultats de l'évaluation, il peut être nécessaire d'ajuster les hyperparamètres tels que le taux d'apprentissage, le nombre d'époques ou la taille des lots (*batch size*) pour optimiser les performances du modèle.

7. **Sauvegarde du Modèle :**

- **Sauvegarde du Modèle Entraîné et du Tokenizer :**
  - Une fois le modèle entraîné, il est important de le sauvegarder ainsi que le tokenizer utilisé. Cela permet une utilisation future sans nécessiter de réentraînement.
- **Exportation des Résultats :**
  - Les résultats et les prédictions du modèle sont exportés pour une analyse plus approfondie et pour permettre des comparaisons avec d'autres modèles ou approches.

5.4 Analyse des Résultats

Après avoir effectué le fine-tuning des modèles CamemBERT et Mistral 7B sur notre corpus annoté en format BIO, nous avons évalué les performances de chaque modèle en

utilisant les métriques de précision, rappel, F1-scoreLes résultats obtenus sont résumés dans les tableaux suivants.

Paramètre	CamemBERT	Mistral 7B
Précision	85.4%	90.1%
Rappel	83.9%	89.5%
F1-Score	84.1%	89.8%

Table 5.1 – Comparaison des performances de CamemBERT et Mistral 7B

5.4.1 Comparaison des Modèles

En comparant les deux modèles, nous observons que Mistral 7B présente des performances légèrement supérieures à celles de CamemBERT sur l'ensemble des métriques, en particulier sur le F1-score et l'exactitude globale. Cette différence peut être attribuée à la capacité de Mistral 7B à traiter des séquences plus longues et à intégrer des informations contextuelles plus larges.

Ces résultats montrent que, bien que les deux modèles soient efficaces pour la tâche de reconnaissance d'entités nommées (NER), Mistral 7B pourrait offrir des avantages supplémentaires pour des applications nécessitant une analyse plus profonde et une meilleure compréhension du contexte.

5.5 Conclusion

Ce chapitre a présenté une étude approfondie des techniques de fine-tuning appliquées aux modèles de traitement automatique du langage naturel (NLP) pour la tâche de reconnaissance d'entités nommées (NER) dans les CV. Nous avons exploré l'utilisation de CamemBERT, un modèle optimisé pour la langue française, ainsi que de Mistral 7B, un modèle de grande envergure (LLM) capable de traiter des corpus textuels vastes et diversifiés.

Le processus de fine-tuning a été détaillé, incluant la préparation des données annotées en format BIO, l'entraînement des modèles, et l'évaluation de leurs performances sur des ensembles de données distincts. Les résultats obtenus ont montré que Mistral 7B, bien que plus complexe, offrait des performances légèrement supérieures à celles de CamemBERT, notamment en termes de F1-score et d'exactitude globale.

L'analyse comparative des modèles a révélé que, bien que CamemBERT soit très performant pour des tâches nécessitant une compréhension fine de la langue française, Mistral 7B apporte une flexibilité et une puissance supplémentaires pour traiter des tâches NLP plus complexes et contextuelles.

Chapitre 6

Conclusion générale

Nous avons exploré des approches novatrices pour le traitement et l'analyse automatique des CV en combinant des techniques avancées de traitement automatique du langage naturel (NLP) et de vision par ordinateur. Notre objectif principal était de développer un système robuste capable d'extraire et de structurer efficacement les informations clés des CV, afin de faciliter le travail des départements des ressources humaines et d'améliorer l'efficacité du traitement des candidatures.

Notre exploration a débuté par une analyse des techniques de vision par ordinateur, en mettant particulièrement l'accent sur les modèles de détection d'objets comme Faster R-CNN, ainsi que sur les outils de reconnaissance optique de caractères (OCR) tels que EasyOCR. Ces technologies ont été intégrées pour extraire avec précision le texte des différentes sections des CV, préparant ainsi les données pour les étapes suivantes de traitement NLP.

Nous avons ensuite optimisé deux modèles NLP avancés, CamemBERT et Mistral 7B, pour la tâche de reconnaissance d'entités nommées (NER). CamemBERT, spécialement adapté pour la langue française, et Mistral 7B, un LLM, ont été fine-tunés sur un corpus de CV annotés. Nos résultats ont démontré la supériorité de Mistral 7B en termes de précision et de F1-score, soulignant ainsi l'importance d'utiliser des modèles adaptés aux spécificités linguistiques.

Les implications de ce travail dans le domaine des ressources humaines, offre une automatisation efficace de l'extraction d'informations des CV, ce qui permet non seulement de gagner du temps mais aussi d'améliorer la précision et la cohérence des données traitées. Ces modèles peuvent être intégrés dans des systèmes de gestion des candidatures pour automatiser le tri représentant une solution efficace et évolutive pour les entreprises.

En perspective, nous envisageons d'explorer l'intégration de modèles de vision par ordinateur plus avancés pour améliorer la segmentation et l'extraction des sections des CV, ainsi que de développer des techniques de fine-tuning encore plus sophistiquées pour optimiser les performances des modèles NLP. De plus, étendre cette recherche à d'autres contextes culturels pourrait permettre de développer des modèles encore plus polyvalents et adaptés aux besoins diversifiés des entreprises modernes.

Résumé

Titre : Apprentissage profond et automatisation d'extraction d'information des Cvs.

Résumé : Dans notre projet d'automatisation de l'extraction d'informations à partir de curriculum vitae (CV), nous avons utilisé une gamme de techniques d'analyse et de traitement des données. Cette dissertation décrit comment nous avons appliqué ces méthodes et technologies pour atteindre nos objectifs.

Tout d'abord, nous avons mis l'accent sur l'analyse de la structure des CV afin de mieux comprendre leur organisation et de faciliter l'extraction ciblée des données. Nous avons employé des techniques de détection d'objets, y compris Faster R-CNN et Mask R-CNN, pour identifier visuellement divers éléments structurels des CV. Des algorithmes d'analyse sémantique nous ont aidés à comprendre la hiérarchie et l'organisation des informations dans les CV, permettant une meilleure contextualisation des données extraites.

Compte tenu des formats divers des CV (image, PDF, etc.), la reconnaissance optique de caractères (OCR) était cruciale pour extraire le texte brut de ces documents. Nous avons principalement utilisé Tesseract, une bibliothèque OCR open-source, et également évalué EasyOCR et PaddleOCR pour comparer leurs performances et choisir la plus adaptée à notre projet.

Le traitement du langage naturel (NLP) a joué un rôle central, nous permettant de saisir le sens et le contexte des informations contenues dans les CV. Les techniques d'analyse sémantique nous ont aidés à comprendre la pertinence des informations, tandis que la reconnaissance d'entités nommées (NER) a permis d'identifier avec précision des éléments clés tels que les noms, les entreprises et les diplômes.

Nous avons intégré ces analyses et technologies dans un flux de travail global combinant l'analyse de la structure, l'OCR, le NLP et l'apprentissage profond pour traiter efficacement les CV. Cette dissertation met également en évidence les domaines d'amélioration pour les travaux futurs.

Mots cles : Analyse Document, OCR, NLP, Faster RCNN, Transformers, Extraction d'informations, NER, Fine Tuning, Apprentissage profond, Reconnaissance de structures

Abstract

Title : Deep learning and automation of information extraction from CVs.

Summary : In our project to automate information extraction from curriculum vitae (CVs), we utilized a range of data analysis and processing techniques. This dissertation describes how we applied these methods and technologies to achieve our objectives.

First, we focused on analyzing the structure of CVs to better understand their organization and facilitate targeted data extraction. Object detection techniques, including Faster R-CNN and Mask R-CNN, were employed to visually identify various structural elements of CVs. Semantic analysis algorithms helped us understand the hierarchy and organization of information within CVs, enabling better contextualization of the extracted data.

Given the diverse formats of CVs (image, PDF, etc.), Optical Character Recognition (OCR) was crucial for extracting raw text from these documents. We primarily used Tesseract, an open-source OCR library, and also evaluated EasyOCR and PaddleOCR to compare their performance and select the most suitable one for our project.

Natural language processing (NLP) played a central role, allowing us to grasp the meaning and context of the information in CVs. Semantic analysis techniques helped us comprehend the information's relevance, while Named Entity Recognition accurately identified key elements such as names, companies, and degrees.

We experimented with various pre-trained language models, including CamemBERT and Mistral, and fine-tuned them for our specific task of CV information extraction. This fine-tuning on annotated CV data significantly enhanced the models' performance.

We integrated these analyses and technologies into a comprehensive workflow combining structure analysis, OCR, NLP, and deep learning to effectively process CVs. This dissertation also highlights areas for improvement for future work.

Keywords : Document Analysis, OCR, NLP, Faster RCNN, Transformers, Information Extraction, NER, Fine Tuning, Deep Learning, Structure Recognition

Bibliographie

[1] L. Guo, Y. Wang et H. Chen. "Automated CV Information Extraction : Challenges and Opportunities in the Digital Age". In : *Journal of Intelligent Information Systems* 60.2 (2023), p. 345-367. doi : 10.1007/s10844-022-00674-z.

[2] Q. Zhang, X. Li et Y. Wu. "Advanced OCR Techniques for Complex Document Processing : A Case Study on Resume Analysis". In : *Pattern Recognition Letters* 158 (2022), p. 50-57. doi : 10.1016/j.patrec.2022.04.031.

[3] S. Dupont, C. Martin et F. Lefevre. "Leveraging Language-Specific Pre-trained Models for Enhanced Information Extraction from French Documents". In : *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Toronto, Canada : Association for Computational Linguistics, 2024, p. 1123-1135. doi : 10.18653/v1/2024.acl-long.96.

[4] P. Kumar. *The Evolution of Deep Learning : A Comprehensive Timeline*. Available at : https ://dataspacein.com/the-evolution-of-deep-learning-a-comprehensive-timeline/, Accessed : 2024-09-29. 2023.

[5] E. Alpaydin et al. *Pattern Classification*. John Wiley & Sons, 2016.

[6] T. Hastie, R. Tibshirani et J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.

[7] M. Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In : *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. 1996, p. 226-231.

[8] J. Manyika et al. *Big data : The next frontier for innovation, competition, and productivity*. Rapp. tech. McKinsey Global Institute, 2011.

[9] P. Buneman. "Semistructured data". In : *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*. 1997, p. 117-121.

[10] D. A. Freedman. *Statistical models : theory and practice*. Cambridge University Press, 2009.

[11] D. W. Hosmer Jr., S. Lemeshow et R. X. Sturdivant. *Applied logistic regression*. T. 398. John Wiley & Sons, 2000.

[12] *Qu'est-ce que le deep learning ?* Available at : https ://mailchimp.com/fr/resources/deep-learning/ Accessed : 2024-09-29. 2023.

[13] Y. LeCun et al. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[14] A. Krizhevsky, I. Sutskever et G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In : *Advances in Neural Information Processing Systems*. 2012, p. 1097-1105.

[15] P. Pedamkar. *DNN Neural Network*. Available at : https ://www.educba.com/dnn-neural-network/ Accessed : 2024-09-29. 2023.

[16] Y. LeCun, Y. Bengio et G. Hinton. “Deep learning”. In : *Nature* 521.7553 (2015), p. 436-444.

[17] I. Goodfellow, Y. Bengio et A. Courville. *Deep learning*. MIT Press, 2016.

[18] M. K. Gurucharan. *Basic CNN Architecture : Explaining 5 Layers of Convolutional Neural Network*. Last updated : 27th Jul, 2022. 2022. URL : https://www.upgrad.com/blog/basic-cnn-architecture/.

[19] *When to use Recurrent Neural Networks (RNN) ?* Accessed : 2024-09-29. URL : https://iq.opengenus.org/when-to-use-recurrent-neural-networks-rnn/.

[20] S. Hesaraki. *Long Short-Term Memory (LSTM) : Architecture and Applications*. Available at : https ://medium.com/@saba99/long-short-term-memory-lstm-fffc5eaebfdc Last updated : Oct 27, 2023. 2023.

[21] Sinno Jialin Pan et Qiang Yang. “A Survey on Transfer Learning”. In : *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), p. 1345-1359. DOI : 10.1109/TKDE.2009.191.

[22] Lisa Torrey et Jude Shavlik. “Transfer Learning”. In : *Handbook of Research on Machine Learning Applications and Trends : Algorithms, Methods, and Techniques*. IGI Global, 2010, p. 242-264. DOI : 10.4018/978-1-60566-766-9.ch011.

[23] Jason Yosinski et al. “How transferable are features in deep neural networks ?” In : *Advances in Neural Information Processing Systems*. 2014, p. 3320-3328.

[24] Simon Kornblith, Jonathon Shlens et Quoc V. Le. “Do Better ImageNet Models Transfer Better ?” In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, p. 2661-2671.

[25] A. Krizhevsky, I. Sutskever et G.E. Hinton. “Imagenet classification with deep convolutional neural networks”. In : *Advances in Neural Information Processing Systems*. 2012, p. 1097-1105.

[26] X. Xu et al. *Efficient fine-tuning of pretrained models for text classification*. 2019. arXiv : 1908.04812 [cs.CL].

[27] McKinsey & Company. “The future of work : Rethinking talent in the digital age”. In : *McKinsey & Company Report* (2018).

[28] Capterra. *2020 Recruiting Trends Report*. Report. 2020.

[29] Boston Consulting Group. *Global Talent Shortage Survey*. Report. 2018.

[30] Laboratoire de Recherche en Informatique, Université Paris-Saclay. "Artificial Intelligence for CV Parsing : Opportunities and Challenges". In : *LRI Research Paper* (2021).

[31] S. Sarawagi. "Information extraction". In : *Foundations and Trends in Databases* 1.3 (2008), p. 261-377.

[32] E. Cortez et al. "A flexible approach for extracting metadata from bibliographic citations". In : *Journal of the American Society for Information Science and Technology* 61.6 (2010), p. 1144-1165.

[33] H. Huang, J. Xu et K.-F. Wong. "A deep learning approach for resume parsing". In : *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM. 2015, p. 1201-1204.

[34] G. Lample et al. "Neural architectures for named entity recognition". In : *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, p. 1487-1497.

[35] L. Zhang et J. Zhao. "Resume information extraction based on BERT-BiLSTM-CRF". In : *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE. 2020, p. 1-6.

[36] R. C. Gonzalez et R. E. Woods. *Digital Image Processing*. Pearson, 2018.

[37] P. Viola et M. Jones. "Rapid object detection using a boosted cascade of simple features". In : *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. T. 1. IEEE. 2001, p. I-I.

[38] K. He et al. "Mask R-CNN". In : *Proceedings of the IEEE International Conference on Computer Vision*. 2017, p. 2961-2969.

[39] R. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In : 2014.

[40] R. Girshick. "Fast R-CNN". In : *Proceedings of the IEEE International Conference on Computer Vision*. 2015, p. 1440-1448.

[41] S. Ananth. *Fast R-CNN for Object Detection*. Available at : https ://towardsdatascience. com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022 Accessed : 2024-09-29. 2019.

[42] S. Ren et al. "Faster R-CNN : Towards real-time object detection with region proposal networks". In : 2015.

[43] A. Khazri. *Faster R-CNN Object Detection*. Available at : https ://towardsdatascience.com /faster-rcnn-object-detection-f865e5ed7fc4 Accessed : 2024-09-29. 2019.

[44] H. Devbhankar. *Instance Segmentation with Mask R-CNN*. Available at : https ://towards datascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b Accessed : 2024-09-29. 2020.

[45] L. Martin et al. “CamemBERT : a Tasty French Language Model”. In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[46] B. Muller, N. Godey et R. Castagné. *Hands-on CamemBERT : Une Introduction au Modèle CamemBERT*. Available at : https ://example.com/handson-camembert Accessed : 2024-09-29. Juill. 2022.

[47] A.Q. Jiang et al. *Mistral 7B : The best 7B model to date*. Available at : https ://mistral.ai/news/announcing-mistral-7b/ Accessed : 2024-09-29. Sept. 2023.