Classification automatique à partir d'un flux de documents

Joris Voerman

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE Euclide

Laboratoire Informatique, Image et Interaction (L3i)

THÈSE * présentée par :

Joris VOERMAN

soutenue le : 13 juin 2022

pour obtenir le grade de : **Docteur de l'université de La Rochelle**

Discipline : **Informatique**

Classification automatique à partir d'un flux de documents

JURY :

Jean-Phillipe DOMENGER	Professeur, Université de Bordeaux, Rapporteur
Nicole VINCENT	Professeur, Université Paris Cité, Rapporteur
Jean-Marc OGIER	Professeur, La Rochelle Université, Directeur de thèse
Mickaël COUSTATY	Maître de conférences, La Rochelle Université
Nathalie GIRARD	Maître de conférences, Université de Rennes 1
Mathieu ROCHE	Chercheur HDR, CIRAD, Montpellier
Aurélie JOSEPH	Chef de projet recherche, YOOZ
Vincent POULAIN D'ANDECY	Manager du département de recherche, YOOZ

∗. Thèse CIFRE financée par l'ANRT, associant l'entreprise YOOZ et le laboratoire du L3i de l'université de la Rochelle

Table des matières

1 Introduction **5**
1.1 Contexte 5
1.2 Les Flux de documents 7
1.3 Pourquoi les réseaux neuronaux ? 9

2 État de l'art **13**
2.1 Classification de documents avec des réseaux neuronaux 15
2.1.1 Architectures communes de réseau neuronaux 15
2.1.2 Classification de documents administratifs à partir du texte 19
2.1.3 Classification de documents administratifs à partir de l'image . . 25
2.1.4 Classification multimodale de documents 28
2.1.5 Modèle d'attention 30
2.2 Classification déséquilibrée de documents 32
2.2.1 Renforcement des classes sous-représentées 32
2.2.2 Augmentation des données 33
2.3 Apprentissage avec peu d'exemples 34
2.3.1 Zero-shot learning 34
2.3.2 One/Few-shot learning 37
2.4 Apprentissage incrémental 41
2.4.1 Neural gas 41
2.4.2 Apprentissage profond incrémental 44
2.5 Conclusion sur l'état de l'art 46

3 Protocole d'évaluation et comparatif de l'état de l'art dans le contexte des flux de document **49**
3.1 Corpus 49
3.2 Protocole de test 53
3.2.1 Mesures 53
3.2.2 Tests d'adaptation 56
3.3 Résultats 59
3.3.1 Corpus brut et contexte équilibré 59
3.3.2 Tests d'adaptation 61
3.3.3 Tests sur des données limitées 64
3.4 Conclusion de l'évaluation 68

4 Multimodalité, adaptations et systèmes d'attention dans les cas déséquilibrés 71
4.1 Modèle multimodal 71
4.2 Systèmes d'attention 74
4.2.1 Modèle d'attention textuel 74
4.2.2 Modèle d'attention visuel 74
4.3 Renforcement contre le déséquilibre 77
4.3.1 Fonction de coût pondérée 77
4.3.2 Augmentation des données 78
4.4 Résultats 79
4.4.1 Corpus brut et contexte équilibré 79
4.4.2 Contexte déséquilibré 81
4.5 Conclusion sur la multimodalité, les adaptations et les systèmes d'attention dans un contexte de flux de documents 83

5 Modèle multi-systèmes et cascade 85
5.1 Préambule 85
5.1.1 Multi-systèmes 85
5.2 Cascade de systèmes 86
5.2.1 Principes 86
5.2.2 Processus d'entraînement 87
5.2.3 Processus de décision 90
5.3 Expérimentation 91
5.3.1 Corpus brut et contexte équilibré 92
5.3.2 Adaptation aux flux de documents 93
5.3.3 Analyse par classe 95
5.3.4 Conclusion sur les expériences 98

6 Conclusion 99
6.1 Retour sur le contexte des flux de documents et l'état de l'art 99
6.2 Conclusion sur les expérimentations, les solutions proposées et les apports scientifiques 101
6.3 Axes exploratoires et pistes de réflexion 102

Chapitre 1

Introduction

1.1 Contexte

Les entreprises privées et l'administration publique font face chaque jour à une énorme quantité de documents à traiter. Ces documents sont en provenance de sources diverses, qu'ils viennent de processus internes ou d'entités externes comme des sous-traitants ou d'autres administrations. Le traitement d'une telle quantité de documents requiert beaucoup d'efforts et de main-d'oeuvre si l'on ne fait pas usage d'un système automatique. De plus, ces documents sont en général liés au coeur d'activité d'une entreprise. Ils sont donc de la première importance puisqu'ils peuvent servir à valider des actions ou des décisions ayant des effets internes et/ou externes à celle-ci. La gestion de ces documents devient alors un défi de vitesse et de précision, puisque toute erreur peut avoir de lourdes conséquences en causant des actions ou des décisions erronées, mais également des pertes d'informations possiblement cruciales. Les implications sont diverses allant d'une simple facture impayée à la classification d'une circulaire urgente en tant que courrier destiné à l'archivage. Dans ce contexte, il est nécessaire que le système de traitement automatique dispose d'une haute précision, ou au moins la plus élevée possible. Le traitement des documents est donc un enjeu majeur pour de nombreux acteurs.

La problématique n'est cependant pas la même en fonction de la taille de l'entité considérée. D'un côté, les plus grandes posent le problème de gérer des quantités massives de documents. De l'autre, les entreprises plus petites (voir auto-entreprises) ont certes moins d'éléments à traiter, mais ne disposent pas forcément d'assez de ressources pour opérer une gestion manuelle sans que cela ne pèse sur leur coeur d'activité. Si pour la première situation, l'état de l'art actuel a fourni de nombreuses solutions, certes imparfaites mais efficaces, pour la seconde cela est moins le cas. Notamment car les solutions développées s'adaptent mal aux différents cadres spécifiques associés aux plus petites entreprises. En effet, les solutions présentes sur le marché sont principalement des logiciels conçus et adaptés au contexte particulier de l'entreprise cliente par un travail conjugué avec les développeurs (cf : ITESOFT, ESKER, AUREXUS, Basware, ...). Ces solutions nécessitent donc de gros moyens pour être mises en place et sont de fait hors de propos dans le cas des petites et des moyennes entreprises.

L'entreprise partenaire de cette thèse CIFRE, Yooz, cherche précisément à offrir une solution à cette situation par le biais d'un service web généraliste permettant de traiter automatiquement les flux de documents internes d'une entreprise sans que cela ne néces-

site de trop lourdes adaptations. Cette solution se veut complémentaire de l'existant et bien plus adaptée à celles qui ne peuvent pas trouver leur compte dans des logiciels trop coûteux ou trop complexes à mettre en place.

Yooz est une société basée à Aimargues dans le Gard, centrée sur le traitement automatique des processus d'achats et de facturation pour les entreprises et les cabinets d'expertise comptable. Ce qui comprend la dématérialisation, la reconnaissance et la fouille de document, ainsi que la détection de fraude documentaire. Dans le but de toujours améliorer la qualité et l'efficacité des algorithmes utilisés dans ses solutions, Yooz dispose d'un département de recherche et de développement qui collabore avec le laboratoire L3i de La Rochelle Université depuis 2011, dans des projets sur ces thématiques. Cette collaboration a mené au cours de la thèse à la création d'un labcom : IDEAS, dans laquelle celle-ci s'inscrit pleinement.

De multiples entreprises, à l'instar de Yooz, proposent des solutions proches comme ABBYY, KOFAX et PARASCRIPT. Ces solutions sont nommées "digital mailroom" en anglais [68, 17], ce que l'on pourrait traduire par "gestionnaire numérique de courriers". Ces systèmes se basent sur des combinaisons de méthodes d'apprentissage automatique et de systèmes experts. Les systèmes experts sont les solutions les plus anciennes et offrent de très bonnes performances dans presque toutes les situations. Cependant, il ne s'agit pas de systèmes usant d'apprentissage automatique et ils nécessitent en contrepartie une maintenance permanente. Une maintenance pour laquelle des experts sont de fait nécessaires, ce qui est également le cas pour la partie conception (d'où leur nom). Ces systèmes sont progressivement remplacés par des méthodes d'apprentissage (profond ou non) partout où cela est possible, car elles nécessitent moins de maintenance, bien que leur efficacité dépende de l'utilisation de grosses bases de données étiquetées. C'est la raison pour laquelle les méthodes de machine learning n'ont pu complètement remplacer les systèmes experts. En effet, dans de nombreuses situations, il est impossible de disposer de corpus suffisamment fournis.

L'objectif de cette thèse est donc de proposer un système générique permettant de traiter automatiquement un ensemble hétérogène de documents issus d'entreprise. Le système doit pouvoir classer un document parmi un large éventail de possibilités par l'apprentissage et la reconstruction des liens sémantiques et structurels entre les documents proches (toujours dans l'objectif de remplacer les systèmes experts). L'apprentissage doit se faire sans modèle a priori (ou le moins possible) sur exemple de résultat. Le vocabulaire des documents doit pouvoir être large et multilingue, Yooz ayant une antenne sur le marché américain, le minimum est l'anglais et le français. La méthode proposée doit satisfaire les contraintes industrielles spécifiques :

— Réduire autant que possible le nombre de paramètres, pour assurer l'accessibilité à un public non expert. La clientèle visée par Yooz correspond aux petites et moyennes entreprises, qui ne disposent pas forcément d'experts dans ce domaine.
— Minimiser le nombre erreurs pour assurer la fiabilité des résultats même au prix de plus de rejets. Une erreur est ici considérée comme étant toujours plus coûteuse qu'un rejet, une erreur de classification pouvant être lourde (en occultant un document important ou en partageant un document sensible).
— La dernière contrainte est un faible temps de traitement par document pour pouvoir faire face à la quantité à traiter (de l'ordre de 100 000 à 150 000 documents par jour pour la partie européenne, soit environ deux documents par seconde).

1.2 Les Flux de documents

Maintenant que le contexte est établi, il faut définir plus formellement ce que la solution proposée aura à traiter. Pour modéliser cette entrée, nous nous sommes basé sur la forme de "flux de documents" (document stream) proposée par [17]. Un flux de documents est défini comme étant une séquence de documents très hétérogènes qui apparaissent dans le temps. Le flux est composé de nombreuses classes plus ou moins proches les unes des autres et très inégalement représentées (d'où son hétérogénéité). Cette distribution (voir figure 1.1) inégale des documents prend la forme d'un petit ensemble de classes majoritaires (dites grandes classes) qui constitue le cœur du flux et d'un grand nombre de classes plus petites (dites petites classes). Ces classes mineures peuvent être moyennement représentées (moitié moins que les plus grandes), voire n'être représentées que par un seul exemple. Cela rend l'apprentissage des très petites classes particulièrement complexe. Les flux de documents étant séquentiels, la quantité de documents pour chaque classe va augmenter au fur et à mesure du temps et de l'évolution du flux. Cependant de nouvelles classes peuvent apparaître spontanément et d'autres disparaître. En conséquence, tout corpus d'entraînement que l'on voudrait générer à partir d'un flux de documents héritera de deux propriétés contraignantes : il sera déséquilibré et incomplet.

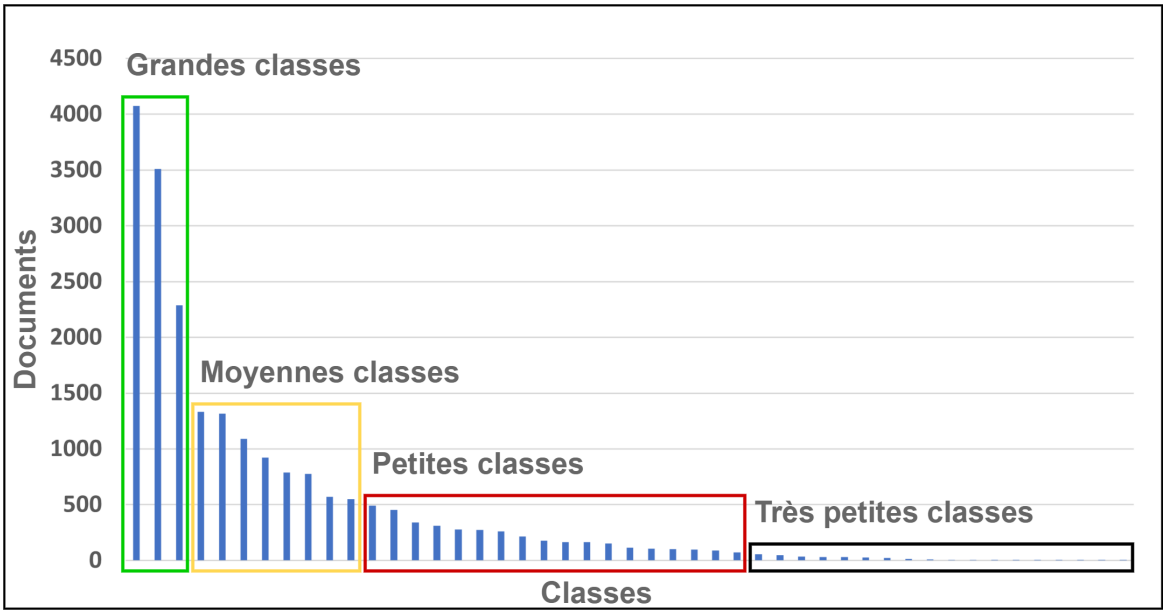

FIGURE 1.1 – Distribution des documents par classe dans un exemple de flux de documents

Le déséquilibre signifie ici l'inégalité du nombre d'exemples de document par classe présente dans l'ensemble d'entraînement. Ce déséquilibre est très fort dans un flux de documents au point où certaines classes manquent d'exemple pour assurer une fiabilité statistique des résultats, comme l'illustre le groupe des très petites classes sur la figure 1.1. Avec seulement un ou deux exemples pour entraîner et pour tester, il devient impossible d'assurer que les résultats obtenus lors d'une expérimentation sont véritablement représentatifs et non dus au hasard. Il est en effet impossible de savoir si les exemples à disposition représentent la diversité interne réelle de la classe. La diversité interne correspond, dans notre contexte, à la différence entre les documents d'une même classe, plus ceux-ci

sont variés plus la diversité interne est élevée. Elle se complète avec la diversité externe, soit la différence entre les classes d'un même corpus.

Dans le cas des flux de documents, l'incomplétude correspond à l'absence d'une partie des classes dans l'ensemble d'entraînement. Cette propriété force le système à devoir apprendre au fur et à mesure les nouvelles classes qui apparaissent dans le flux et qu'il n'a donc pas pu apprendre auparavant. Si le système ne peut pas les apprendre directement, il lui faudra au moins la capacité de les écarter (rejeter) puis d'être mis à jour régulièrement pour intégrer ces nouveautés.

En conséquence, les flux de documents sont des ensembles particulièrement difficiles à classifier surtout pour les méthodes les plus performantes de l'état de l'art dont beaucoup usent de machine à apprentissage statistique comme les réseaux neuronaux. Ces méthodes s'adaptent très mal au flux de documents. En effet, dans une situation de déséquilibre fort, le réseau aura tendance à favoriser les classes les plus représentées dans le corpus d'entraînement, au dépend des classes les moins représentées. Plus le déséquilibre est fort, plus ce favoritisme s'intensifie, rendant le système de plus en plus confiant dans ses décisions et surtout dans ses erreurs. Quant à l'incomplétude, celle-ci nécessite un important réentraînement du réseau à cause du phénomène "d'oubli catastrophique"[53] (ou "catastrophic forgetting". Il s'agit d'une tendance des réseaux neuronaux à oublier une partie des informations apprises précédemment lors de l'introduction d'un nouvel élément).

Pour finir, la caractérisation des documents en eux-mêmes constitue une problématique à part entière. Il y a en effet traditionnellement deux modalités distinctes pour classer des documents : l'image et le texte. Ceci est en partie dû au fait qu'une partie des méthodes utilisées dans le cadre du traitement de document proviennent, à l'origine, d'autres champs de recherches. En l'occurrence, des domaines du traitement de l'image d'une part et du traitement automatique de la langue d'autre part. En conséquence, beaucoup de méthodes n'exploitent qu'une seule modalité. Or, si les documents sont tantôt plus caractérisés par leurs structures, tantôt plus par leur contenu textuel, pour pouvoir traiter le plus grand ensemble possible de documents il devient nécessaire d'être capable de jongler entre les deux. L'être humain fait usage de l'image et du texte pour concevoir, analyser et traiter les documents qu'il utilise, il semble paradoxal de chercher à l'égaler ou à le dépasser en n'utilisant que l'une de ces deux modalités. De fait, avec les documents, de nombreuses situations sont impossibles à résoudre si l'on n'utilise pas la bonne modalité comme l'illustrent les figures 1.2 et 1.3. Les deux modalités se confondent dans notre situation, même si globalement pour la classification de documents administratifs ou d'entreprises, le texte reste l'approche qui semble la plus efficace si l'on devait n'en choisir qu'une. En conséquence de ceci, une approche multimodale semble récemment gagner du terrain au sein de l'état l'art en prenant le pas sur les approches monomodales, même si dans le cas des réseaux neuronaux et de l'apprentissage profond la combinaison des deux modalités reste assez limitée dans sa conception [7].

Les flux de documents sont donc des ensembles difficiles à classer car déséquilibrés et incomplets. Ces propriétés rendent inadaptées les solutions présentes dans l'état de l'art pour la classification de documents. Cette classification est d'ailleurs toujours un verrou scientifique à l'heure actuelle, les méthodes développées spécifiquement pour, offrent des résultats inférieurs à celle du traitement de l'image ou du langage. Les documents sont des éléments complexes à classifier de par leur nature multimodale.

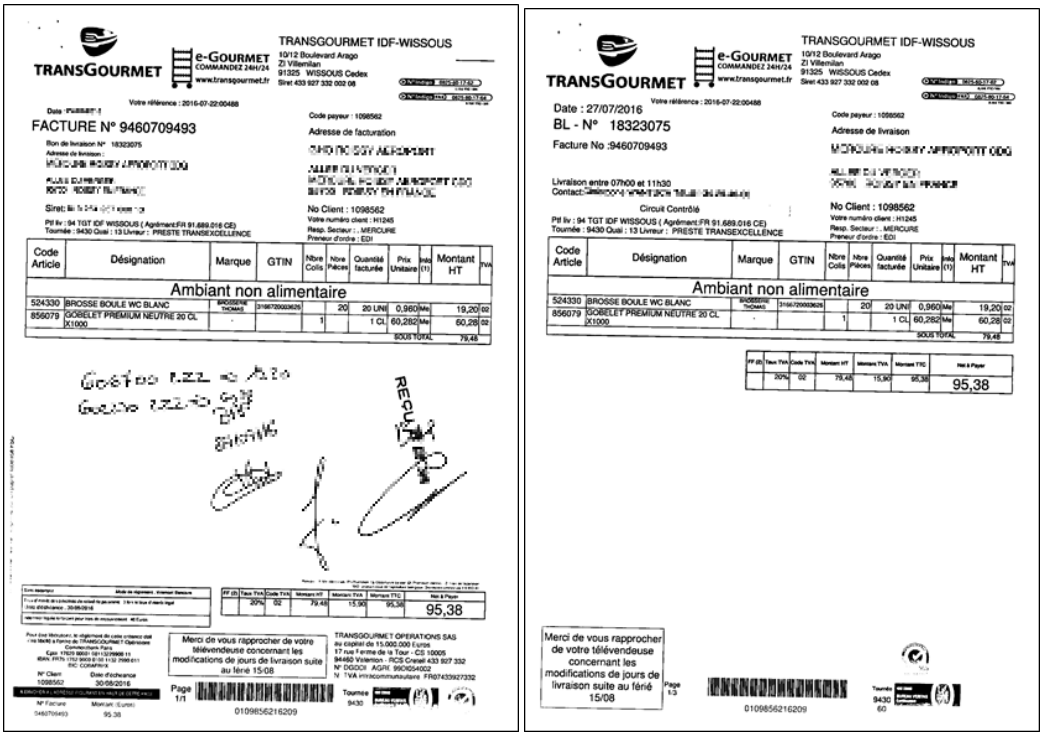

FIGURE 1.2 – Exemple d'un cas difficile pour l'approche image : le premier document est une facture et le second un bon de livraison (Images censurées provenant de la base privée Yooz)

1.3 Pourquoi les réseaux neuronaux ?

Les premières méthodes automatiques permettant la classification et l'extraction depuis un document administratif sont les systèmes dit "experts à base de règles" (Rule-Based Expert System) [58]. Il s'agit de systèmes utilisant une série de règles spécialement conçues pour résoudre une tâche ou un problème précis, dans le but de simuler le processus de décision d'un expert humain [75, 26]. Les types de règles utilisées sont très variés : expressions régulières permettant d'extraire des mots-clés d'un type particulier (comme les entités nommées) ou appartenant à une liste préétablie sur des connaissances à priori, règles utilisant des positions absolues ou relatives pour découper un document en zones précises (très utilisée pour la dématérialisation de formulaire) ou encore règles prenant la forme de formule mathématique pour vérifier une cohérence des valeurs extraites (pour les montants d'une facture), etc. Ces méthodes nécessitent que le document soit lisible et dans le bon sens, ce qui force à les compléter de méthodes de prétraitements pour améliorer autant que possible l'image et le texte extrait du document. Les règles de ces systèmes sont établies par l'homme avant d'être appliquées successivement, il est donc nécessaire de disposer d'une certaine expertise pour pouvoir les établir et de nombreux essais pour les valider [34]. Ces solutions sont toujours utilisées par Yooz pour résoudre plusieurs problèmes de classification et d'extraction d'informations dans les documents.

Les systèmes experts sont des méthodes très performantes et pouvant s'adapter à des situations complexes (représentation très déséquilibrée, documents endommagés, ...). Cependant, ceux-ci nécessitent beaucoup d'intervention humaine pour être conçus, ainsi qu'une forte connaissance à priori du domaine d'application des règles qu'ils utilisent. D'autant plus que, pour s'assurer des performances élevées, les règles doivent être les

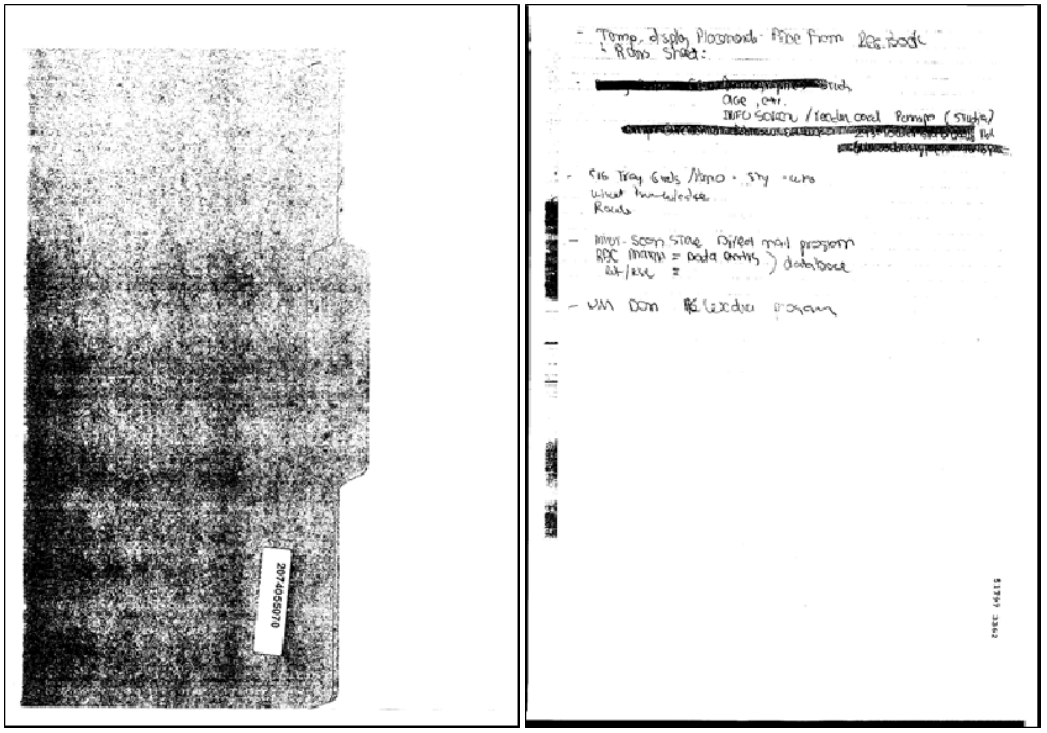

FIGURE 1.3 – Exemple d'un cas difficile pour l'approche texte : le premier document est un séparateur de dossier et le second une note manuscrite. (Image provenant de la base publique RVL-CDIP [27]) Note : Il est impossible pour les OCRs industriels actuels d'extraire convenablement le texte manuscrit, il faut des logiciels spécifiques (OCR ou ICR spécialisés).

plus exhaustives possibles, transformant la conception en un processus laborieux. À noter que pour pouvoir intégrer de nouvelles classes, ces systèmes requièrent de recommencer le même processus de conception (en ajoutant de nouvelles règles et en s'assurant que les anciennes sont toujours valables). Tout cela rend la maintenance très coûteuse, poussant à chercher un moyen de rendre l'apprentissage de ces règles automatiques. En conséquence, de nombreuses solutions sont apparues dans l'état de l'art sous la forme de l'apprentissage machine.

L'apprentissage machine (ou "machine learning") rassemble les algorithmes qui se perfectionnent automatiquement via des expériences réalisées sur un ensemble de données d'entraînement [57]. Parmi ces méthodes, les réseaux neuronaux ont connu un regain d'intérêt depuis plusieurs années, notamment l'apprentissage profond (ou deep learning). Ils affichent de bons résultats dans plusieurs domaines comme la classification d'images (avec les défis autour d'ImageNet et les nombreux réseaux qui les ont remportés [42, 74, 70, 28]), la classification de textes [56] et bien d'autres [2]. Une fois entraînés, ils sont rapides et ne nécessitent aucun paramétrage supplémentaire (beaucoup de paramétrages sont nécessaires avec les réseaux neuronaux mais en amont, donc pas destinés à l'utilisateur final de l'application). Ils peuvent s'adapter à des systèmes de rejet pour être plus précis et nécessitent moins de ressources pour être maintenus que les systèmes experts. En somme, ils remplissent une bonne partie des contraintes industrielles imposées.

Malgré tous ces avantages, les réseaux neuronaux n'ont pas totalement remplacé les systèmes experts dans les applications industrielles autour de la classification de docu-

ments. En effet, les réseau neuronaux semblent avoir des difficultés à s'entraîner dans des contextes déséquilibrés [35] ou avec trop peu d'exemples [78], ce qui risque d'impacter leurs performances. L'étendue de cet impact dans notre contexte reste encore à quantifier. Un autre point négatif pour ces méthodes est la compréhension des décisions et par extension des erreurs, les réseaux étant par bien des aspects des boîtes noires. Le Tableau 1.1 résume la comparaison des avantages et inconvénients des systèmes experts et des réseaux neuronaux.

TABLE 1.1 – Comparaison **avantages**/**inconvénients** entre les réseaux neuronaux et les systèmes experts à base de règles

Critères de comparaison	**Réseaux Neuronaux**	**Systèmes Experts**
Contraintes industrielles		
Vitesse d'exécution	➕ (Dépend du nombre de paramètres, soit la taille du réseau)	➕ (Dépend du nombre de règles et de leur complexité)
Coût Puissance de calcul	➖ (Élevée)	➕ (Faible)
Compréhensibilité des erreurs	➖	➕
Durée du processus de développement (en comparaison)	➕ (Court)	➖➖ (Long)
Connaissance à priori nécessaire ?	➕ (Peu - juste les classes des documents d'entraînement)	➖➖ (Indispensable à un niveau d'expert)
Adaptation au flux		
Résistance au déséquilibre	? (hypothèse : ➖)	➕➕
Résistance à l'incomplétude	? (hypothèse : ➖)	➖ (Limité aux connaissances de l'expert)
Précision sur la majorité du flux (Grandes classes)	➕	➕
Précision sur les petites classes	? (hypothèse : ➖)	➕
Intégration de nouveauté	➖ (Réentraînement, avec besoin d'exemples des nouvelles classes)	➖➖ (Ajout de nouvelles règles et vérification de la validité des anciennes => encore plus long que pour les réseaux neuronaux)

En conclusion, à travers cette thèse nous cherchons une solution permettant de classer des flux de documents déséquilibrés et incomplets dans un contexte industriel. La solution a pour but d'offrir une alternative aux systèmes experts utilisés actuellement, avec des méthodes moins exigeantes au niveau de la maintenance, mais aussi performantes. Les réseaux neuronaux constituent une option attrayante, car ils offrent des résultats très corrects sur les classes fortes. Les adapter au fort déséquilibre et à l'incomplétude des flux de documents, sans que leurs performances ne soient trop impactées, peut constituer une solution efficace pour résoudre la problématique de cette thèse.

Chapitre 2

État de l'art

Dans la littérature, plusieurs domaines de recherches distincts sont en lien avec notre sujet et peuvent proposer des solutions pertinentes pour notre problématique. Elles peuvent être rassemblées en quatre grands axes : la classification de document par réseau neuronal, la classification déséquilibrée, l'apprentissage avec peu d'exemples et l'apprentissage incrémental.

La classification de document avec des réseaux neuronaux se concentre sur les nombreuses méthodes de classification disponibles dans l'état de l'art applicable aux documents, sans en être obligatoirement l'objectif d'origine. Cet axe inclut donc les méthodes de classification d'image, naturelle ou non, et de texte. Une bonne partie des méthodes de classification déjà utilisées pour les documents provient originellement de ces champs de recherches. L'objectif ici est de rechercher les meilleurs candidats afin d'évaluer leurs limites face aux contraintes des flux. Ils pourront alors servir de base de comparaison avec les améliorations proposées.

Une fois les meilleurs candidats rassemblés parmi les méthodes de classification, il devient nécessaire de chercher les contre-mesures aux faiblesses des réseaux dans les cas déséquilibrés (soit avec un nombre d'échantillons très inégal entre les classes) qui ont déjà été développées dans la littérature. Cet axe, appelé ici classification déséquilibrée, englobe l'ensemble de ces solutions développées pour compenser ce défaut. Ces méthodes peuvent convenir à notre problématique si elles restent efficaces même en cas de déséquilibre très fort, comme c'est le cas pour les flux de documents.

Les classes très faiblement représentées sont la partie la plus difficile du flux à gérer pour les méthodes de machine learning. L'apprentissage de classe avec peu d'exemples constitue un domaine à part entière dans la recherche. Les défis de traitement de l'image que sont le "one-shot learning" et le "zero-shot learning" cherchent clairement à résoudre cette problématique. Dans l'hypothèse où ces méthodes parviennent à être appliquées efficacement aux documents, en renforçant les résultats sur les petites classes sans trop perdre au niveau des plus grandes, alors elles pourraient être d'excellentes candidates à étudier. Même si aucune d'entre elles ne parviendrait à remplir ce cahier des charges très exigeant, plusieurs d'entre elles pourraient se montrer inspirantes.

Enfin, il reste les classes incomplètes ou manquantes (celles qui ne seront pas présentes lors de l'entraînement). L'option la plus adaptée dans la littérature pour ce type de problématique est l'apprentissage incrémental. Il s'agit d'un type d'apprentissage où les données arrivent les unes après les autres. Le modèle doit alors être capable d'apprendre de chaque nouvelle donnée, sans perdre de ce qu'il a déjà appris (et donc ne pas refaire un

apprentissage sur toutes les données précédentes) [9]. Il est également possible d'augmenter le nombre de classes au cours d'un apprentissage incrémental, permettant d'ajouter les classes manquantes sans réentraînement. C'est un domaine où il est possible de trouver des méthodes d'apprentissage automatique et des tentatives d'applications sur des réseaux neuronaux, pour les rendre incrémentaux.

Dans la prochaine section nous commencerons par un rappel général concernant les bases sur les réseaux neuronaux, qui sont essentielles d'avoir à l'esprit avant d'aborder la suite.

2.1 Classification de documents avec des réseaux neuronaux

2.1.1 Architectures communes de réseau neuronaux

Les réseaux neuronaux sont des méthodes d'apprentissage statistique, à partir d'exemples de résultats, originellement inspirées du cerveau humain (neurones biologiques) [31]. La première partie de la Figure 2.1, montre le schéma basique d'un neurone j : chaque entrée x_n est multipliée avec le poids correspondant w_{nj}. L'ensemble des entrées est ensuite additionnée par la fonction de transfert de sorte à ce que $net_j = \sum x_n \times w_{nj}$. Enfin la fonction d'activation du neurone (deuxième partie de la Figure 2.1) est appliquée afin que $o_j = \varphi(net_j)$. Un neurone se présente donc sous la forme d'une somme d'entrées pondérées par des poids appris, suivie d'une fonction d'activation.

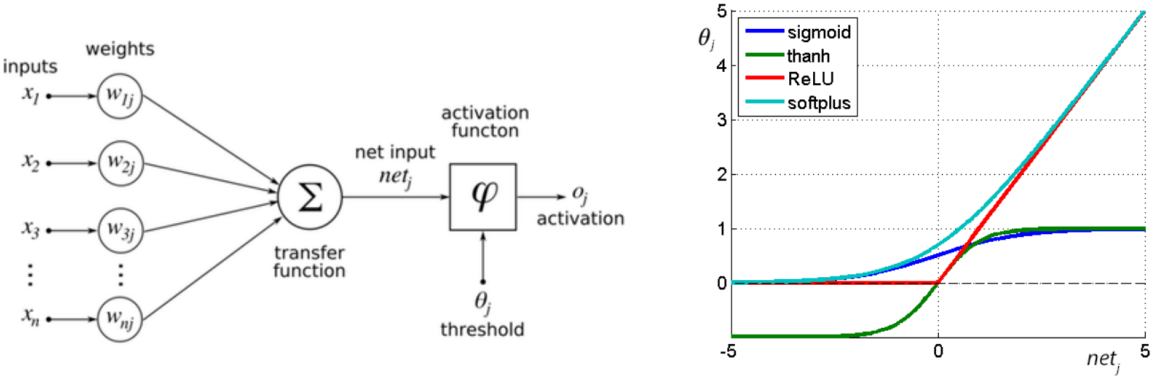

FIGURE 2.1 – Schéma basique d'une neurone à gauche et quatre des fonctions d'activation les plus utilisées à droite [52]

Les neurones sont assemblés sous forme de couches qui se succèdent pour constituer un réseau [52]. Les entrées d'une couche correspondent à la sortie de celle qui la précède. La toute première du réseau, dite couche d'entrée, n'est autre que le vecteur représentant la donnée à analyser. La dernière couche du réseau est dite couche de sortie et toutes les couches entre elle et l'entrée sont dites couches cachées. Les couches basiques utilisant l'architecture de la Figure 2.2 sont appelées des couches "entièrement connectées" (Fully connected layers) ou "couche dense". Les réseaux utilisant uniquement cette architecture de neurone simple avec un seul sens sont appelés "Feed Forward". Notez que la notion d'apprentissage profond (deep learning), très présente dans l'état de l'art des réseaux neuronaux, signifie simplement qu'il y a beaucoup de couches cachées dans le réseau.

L'entraînement d'un réseau neuronal comme le "Feed Forward" [65] se fait au moyen d'une boucle permettant d'apprendre sur l'ensemble des exemples d'entraînement plusieurs fois. Un tour de la boucle, soit un entraînement une fois sur le corpus au complet est appelé une époque (epoch). Une époque est également subdivisée en étapes (steps) correspondant à un processus d'entraînement sur un lot (batch) de plusieurs exemples. Ce processus d'entraînement fonctionne par une comparaison entre le résultat renvoyé par le modèle et la vérité terrain d'un exemple (réponse attendue, dans notre cas la vraie classe du document établie par l'homme lors de la création du corpus). La comparaison est opérée par une fonction de coût (loss function) qui renvoie donc le coût (loss), soit la valeur chiffrée de l'erreur calculée entre le résultat et la vérité terrain. Ce coût est ensuite transmis à une fonction d'optimisation (la plus commune est la descente de gradient stochastique

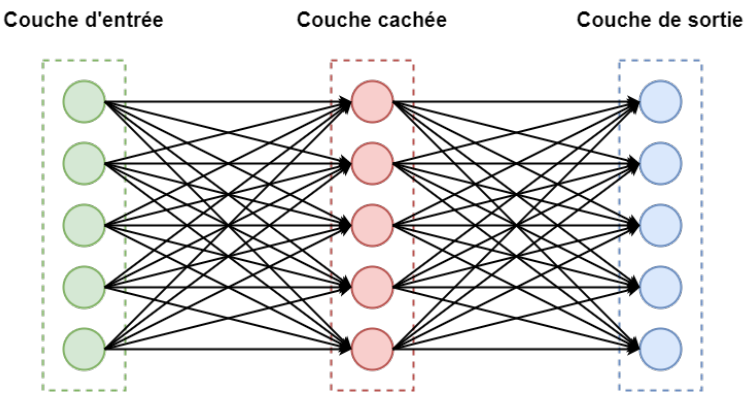

FIGURE 2.2 – Réseau neuronal simple "Feed Forward"

(SGD) [37]). Cette nouvelle fonction calcule alors à partir du coût, un gradient qui est appliqué par rétro-propagation à l'ensemble du réseau pour corriger l'erreur observée en sortie. La correction s'applique sur les paramètres du réseau (majoritairement des poids) pour que la prochaine fois qu'il rencontre cet exemple, la sortie se rapproche de la vérité terrain. Pour permettre une convergence des paramètres du réseau entre les exemples, cette correction est pondérée par un taux d'apprentissage (qui ne conserve qu'un pourcentage de la correction). L'organisation sous forme de lots permet d'ailleurs de lisser statistiquement les coûts des exemples entre eux et donc de faciliter encore la convergence du réseau.

Les réseaux basiques "Feed Forward" ont une structure adaptée à une entrée vectorielle. Selon le type d'entrée à analyser, d'autres types d'architectures plus complexes sont plus adéquates. Les architectures plus communes sont : les réseaux récurrents (RNN) [24] et les réseaux convolutifs (CNN) [59]. Les réseaux récurrents sont plus adaptés aux séquences. Ils ressemblent beaucoup aux "Feed Forward", mais ajoutent à l'entrée (élément courant de la séquence) la sortie du réseau obtenu avec l'élément précédent de la même séquence (voir Figure 2.3). Cette adaptation de la structure du réseau offre la possibilité de conserver des informations entre les éléments et donc de contextualiser un élément au sein de sa séquence.

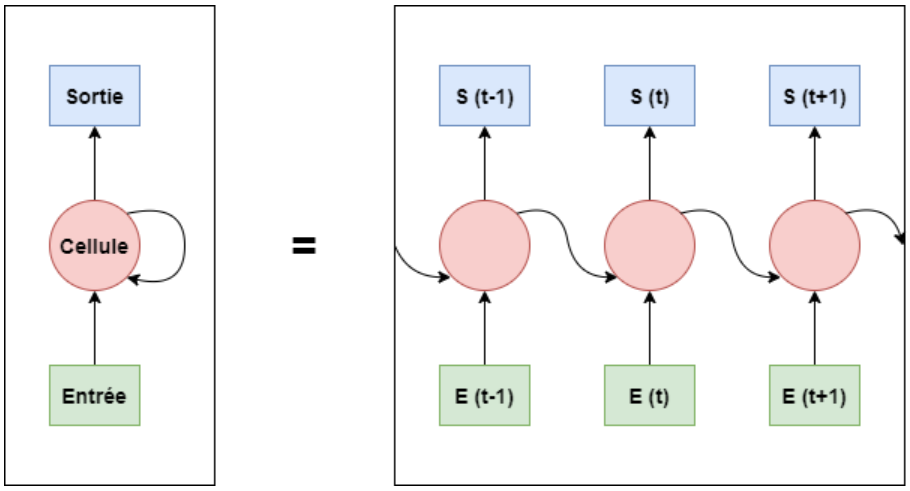

FIGURE 2.3 – Structure basique d'un réseau récurrent

La seconde architecture la plus commune dans l'état de l'art est le CNN. Les CNN sont plutôt conçus pour l'analyse de matrices en permettant l'extraction de caractéristiques multidimensionnelles (là où les réseaux "Feed Forward", sont plus adaptés à une entrée uni-dimensionnelle par la forme de leurs couches). La particularité des CNN est leur utilisation de couches de convolution, où les neurones classiques sont remplacés par une série de plusieurs filtres de convolution de taille fixe (et égale au sein d'une même couche). La sortie de la couche correspond aux résultats de la convolution de chaque filtre sur l'entrée. L'apprentissage se fait sur les poids des filtres. Dans le cas de la classification, les CNN se terminent en général par un bloc de couche dense permettant de transformer la sortie des couches de convolution en un vecteur one-hot (vecteur dont la taille est égale au nombre de classes, l'index de la valeur la plus élevée du vecteur donne le numéro de la classe).

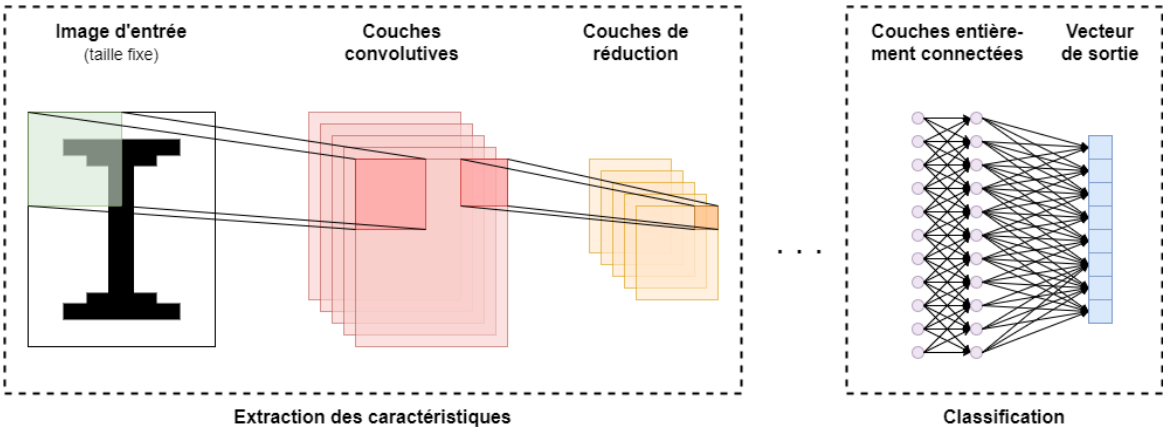

FIGURE 2.4 – Structure basique d'un réseau convolutif

Les réseaux convolutifs qui se montrent comme étant les plus efficaces pour de la classification sont des versions très profondes comme AlexNet [42] qui a dix couches cachées. Cependant, l'entraînement des réseaux neuronaux profonds pose plusieurs problèmes [22] comme la disparition du gradient (vanishing gradient), l'explosion du gradient (exploding gradient) ou la dégradation liée à la profondeur du réseau (degradation / accuracy saturation). La disparition du gradient se manifeste par une réduction exponentielle du gradient due à l'application successive de fonctions d'activation comme la tangente hyperbolique. "L'explosion" est l'effet inverse lié à une forte augmentation de la valeur des paramètres entraînés du réseau. Ces deux problèmes empêchent le réseau de s'entraîner en perturbant le calcul du gradient de correction (voir section 1.3). Il est possible de résoudre ces problèmes par l'ajout d'une couche de normalisation entre les blocs de convolutions (ou groupe de couches pour les réseaux non convolutifs) et par l'utilisation de fonction d'activation ReLU (Unité Linéaire Rectifiée) dans les neurones. Le troisième problème est une saturation des performances à partir d'une certaine profondeur suivie d'une dégradation si le réseau devient plus profond. ResNet [28] propose une solution à ce problème par l'introduction de blocs d'apprentissage résiduels. Ce type de bloc est composé de deux couches de convolutions avec une fonction d'activation ReLU, suivie d'une addition entre la sortie de la dernière et l'entrée du bloc (voir Figure 2.5).

Il existe également un autre problème qui peut apparaître lors de l'entraînement d'un réseau neuronal : le sur-apprentissage (overfitting). Le sur-apprentissage est commun à

tous les systèmes à entraînement statistique dont font partie les réseaux neuronaux. Il est caractérisé par un ajustement du réseau prenant trop en compte les particularités du jeu de données utilisé pour son entraînement. Ce phénomène induit une incapacité du système à généraliser les connaissances acquises durant l'entraînement et donc une perte de performance globale lors des tests sur d'autres données. Pour contrecarrer ce problème, la méthode la plus utilisée est le "dropout" [73]. Il s'agit d'une pratique consistant à éliminer les neurones redondants dans les couches denses.

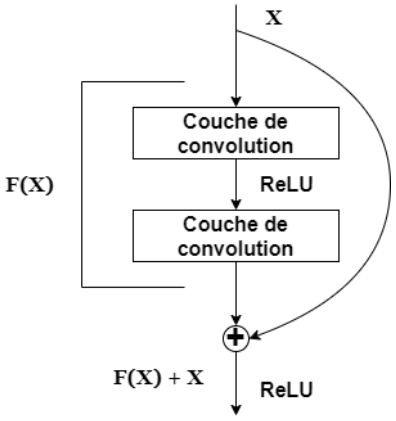

FIGURE 2.5 – Architecture d'un bloc résiduel de ResNet [28]

Maintenant que les bases sont définies, les sections suivantes entreront plus en détail dans la description des méthodes de l'état de l'art concernant la classification via des réseaux neuronaux applicables aux documents administratifs. Les méthodes seront regroupées selon la modalité ou les modalités qu'elles utilisent (texte, image ou les deux).

2.1.2 Classification de documents administratifs à partir du texte

Les réseaux neuronaux textuels sont utilisés pour de nombreuses tâches du domaine du traitement automatique de la langue (TAL) comme la traduction [77], la fouille, l'analyse de phrases [38] et, ce qui nous intéresse ici, la classification de texte [44] et plus spécifiquement de documents [16]. Les architectures des réseaux conçues pour une de ces tâches peuvent parfaitement être utilisées pour une autre (moyennant quelques adaptations si nécessaire). Les innovations apportées par les différentes méthodes portent sur le prétraitement du texte, les cellules du réseau et/ou l'architecture globale.

Pour commencer, l'entrée d'un réseau est toujours une matrice numérique. En conséquence, cela pose la question de l'encodage du texte qui constitue alors le premier élément de tout réseau textuel. L'encodage le plus simple avec un nombre pour un mot (exemple : l'encodage ASCII) n'est clairement pas la méthode la plus efficace et montre de nombreuses limites. La plus importante est qu'elle ne permet pas de retranscrire pour la machine le sens associé aux mots. Pour pallier cette importante perte d'information, l'option retenue est plutôt d'appliquer une méthode dite "d'encapsulation de mots" (word embedding). L'encapsulation permet de transformer un mot, un groupement ou des fragments de mots en vecteurs qui permettent de retranscrire autant que possible le sens du mot encapsulé dans son contexte. Il s'agit d'un champ de recherche prolifique depuis quelques années et qui a amené de nombreuses méthodes. Les plus connues sont Word2Vec [54], GloVe [61], Fasttext [36] et plus récemment BERT [19]. BERT semble se démarquer comme la plus efficace des quatre, nos propres tests vont dans ce sens également. Cependant, cela reste difficile à affirmer faute de comparaisons récentes l'incluant.

Dans leur fonctionnement, les modèles d'encapsulation de mots se basent sur des méthodes d'extraction de caractéristiques textuelles plus anciennes : les sacs de mots (Bag of word) et les Skip-Gram pour Word2Vec, ainsi que les matrices de cooccurrences pour GloVe. Ces méthodes d'extraction sont couplées avec des réseaux neuronaux pour permettre un renforcement par apprentissage. Fasttext est une méthode un peu à part car elle cherche avant tout à compresser le texte durant le processus d'encapsulation (BERT sera présenté plus en détail un peu plus loin).

Pour entraîner les modèles d'encapsulation, il faut d'immenses corpus de textes qui n'ont heureusement pas besoin d'être annotés, car ils s'entraînent en utilisant la récurrence des mots et de leurs contextes au sein du texte (Si le texte est correctement écrit, il constitue déjà la vérité terrain). Les corpus rassemblant tous les articles de Wikipédia (uniquement les paragraphes de textes) sont fréquemment utilisés à cette fin car ils sont volumineux (plus de 2 milliards de mots pour le Wikipédia anglophone) et facilement accessibles. Le corpus francophone est moins fourni mais reste suffisant pour entraîner un modèle de langue efficace avec ces méthodes.

Pour entrer plus dans le détail, Word2Vec [54] propose deux types d'architectures conçus pour apprendre un modèle de langue avec une version améliorée des NNLM [8]. Le modèle de langue entraîné permet de convertir un mot en vecteur en fonction de son contexte, avec pour objectif que les mots proches voient leurs vecteurs l'être également. La première architecture est un modèle de sacs de mots continue (Continuous Bag-of-Words), où l'ensemble des mots du contexte de celui considéré sont projetés dans la même position en moyennant leurs vecteurs (voir la première partie de la Figure 2.6). En conséquence, l'ordre des mots n'a pas d'importance pour le CBOW (d'où son nom).

Celui-ci s'entraîne en cherchant à prédire un mot à partir de son contexte. La seconde architecture, appelée Skip-gram, est la même que pour le CBOW mais inversée. L'objectif devient alors de retrouver le contexte à partir du mot, comme le montre la deuxième partie de la Figure 2.6. Cette version a la particularité de prendre en considération l'ordre des mots et de maximiser le rapprochement entre deux mots qui apparaissent fréquemment dans les mêmes phases. Le modèle Skip-gram se montre comme étant le plus intéressant des deux [55].

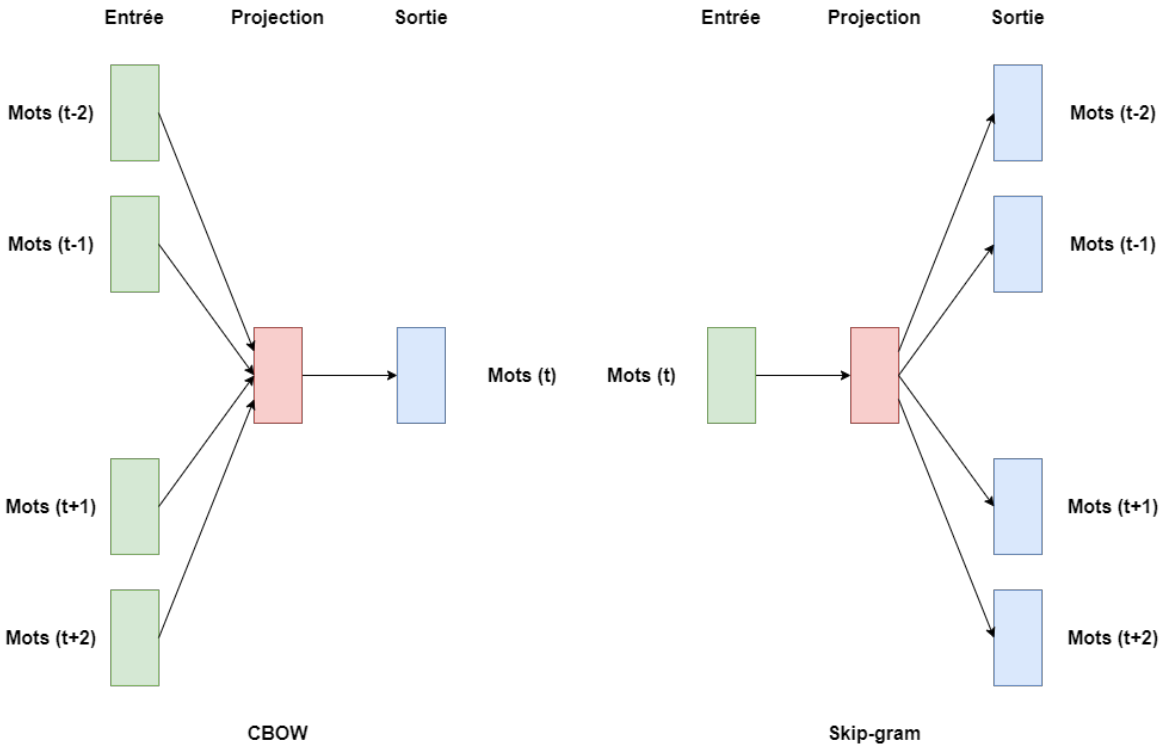

FIGURE 2.6 – Architecture des modèles CBOW et Skip-gram de Word2Vec [54]

BERT [19] utilise lui un encodeur bidirectionnel multi-couches originellement destiné à la traduction automatique [77]. Cet encodeur est renforcé à plusieurs niveaux par des systèmes d'attention (voir Section 2.1.5) pour faciliter le rapprochement entre des mots éloignés dans la même phrase. De plus, les mots et phrases fournis en entrée de l'encodeur sont réarrangés par un système appelé WordPiece [81] issus également de la traduction automatique. Ce système permet d'associer des paires de phrases liées en une seule séquence avec l'introduction de marqueurs de séparation. Il permet aussi de découper les mots complexes en multiples fragments. La fragmentation des mots est utilisée pour compenser la rareté de certains mots en les subdivisant en racines pensées pour être plus communes (typiquement une séparation racine verbale et terminaison "ing" en anglais).

Une fois le texte transformé pour pouvoir servir d'entrée à un réseau, se pose la question de l'architecture du réseau lui-même. Ces architectures de réseaux sont majoritairement récurrentes, pour conserver autant que possible l'information séquentielle des phrases. La récurrence peut également être utilisée dans les deux sens, ils sont alors dits réseaux récurrents bidirectionnels (voir Figure 2.7), à l'instar de celui utilisé par BERT.

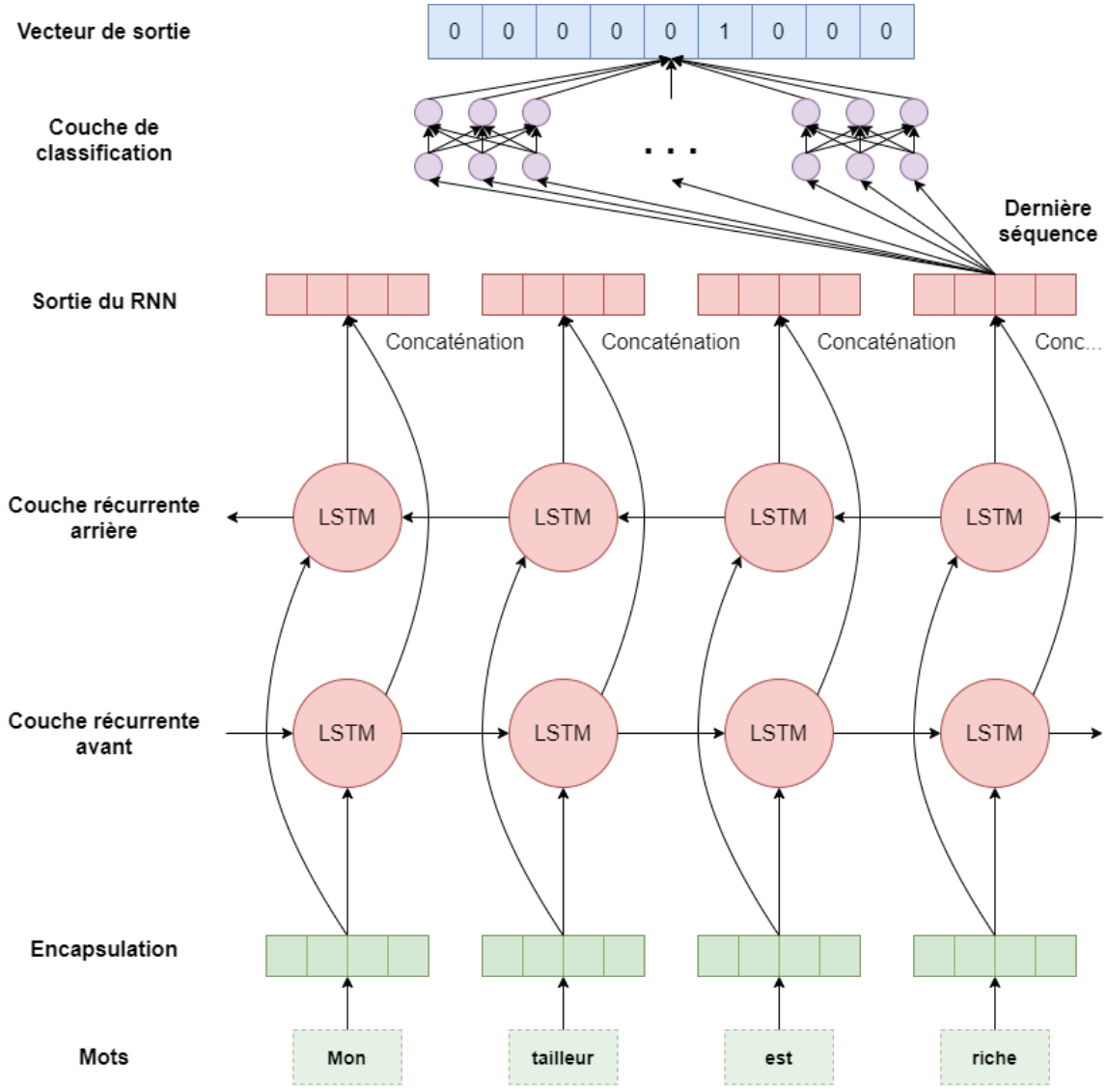

FIGURE 2.7 – Architecture d'un réseau bidirectionnel

Pour renforcer ce type de réseaux, il existe des neurones plus spécifiques et mieux adaptés : les LSTM [29] et les GRU [14]. Ces cellules sont conçues pour compenser l'incapacité des réseaux récurrents à conserver des informations sur le long terme pour mieux apprendre les connexions entre des éléments distants au sein de la séquence.

Les LSTM [29], ou "Long Short-Term Memory", se caractérisent par une complexification des cellules neuronales classiques (comme décrites dans la section 1.3) liée à l'ajout d'un système de mémoire interne à la cellule. Cette mémoire permet de stocker une information (entrée de la cellule) précédemment rencontrée dans la séquence aussi longtemps que nécessaire ou plutôt jusqu'à ce qu'elle soit remplacée par une autre. En effet, la mémoire interne ne permet de stocker qu'une seule information à la fois ce qui induit un mécanisme d'oubli. La structure interne d'une LSTM suit le deuxième schéma de la figure 2.8 où apparaissent plusieurs portes. Ces portes servent à réguler l'utilisation de la mémoire de la LSTM et sont au nombre de trois pour les versions les plus communes.

Les GRU [14], ou "Gated Recurrent Unit", ressemblent aux LSTM bien qu'elles aient

une porte de moins. Elles sont connues pour être plus rapides à calculer que leurs homologues, tout en restant aussi performantes dans plusieurs domaines d'application. En somme, elles constituent une alternative plus rapide mais moins performante.

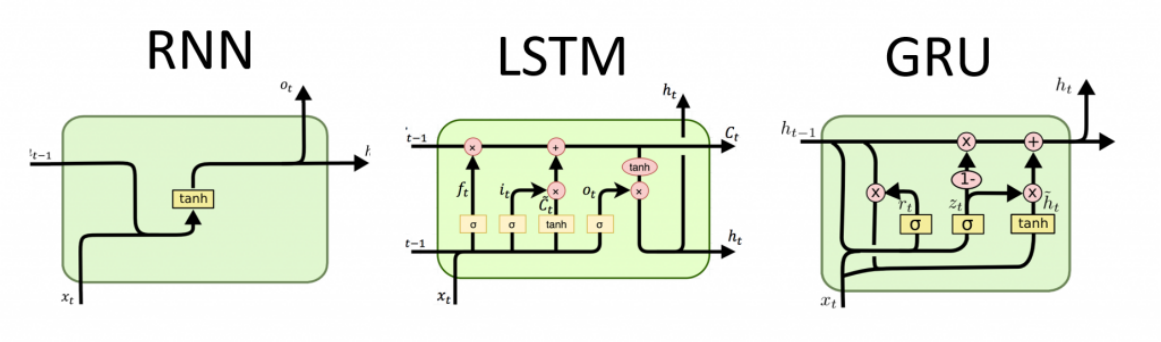

FIGURE 2.8 – L'intérieur des trois types de cellules récurrentes les plus courantes [76]

L'utilisation de CNN (donc sans récurrence) est également une option qui a montré son efficacité dans plusieurs méthodes de l'état de l'art [38, 85]. La première propose de représenter les phrases sous forme de matrice de taille n x k, où n est le nombre de mots par phrase (fixe) et k la taille du vecteur représentant le mot. Les vecteurs des mots sont concaténés successivement dans l'ordre d'apparition pour former la matrice représentant ainsi la phrase. Ensuite, l'architecture se déploie en suivant celle d'un CNN classique avec un bloc de convolution ayant de multiples filtres. Ces filtres sont toujours de longueur k, s'appliquant donc comme une fenêtre coulissante. Ce qui varie est la hauteur, soit le nombre de mots considéré dans une fenêtre. L'architecture suit la figure 2.9.

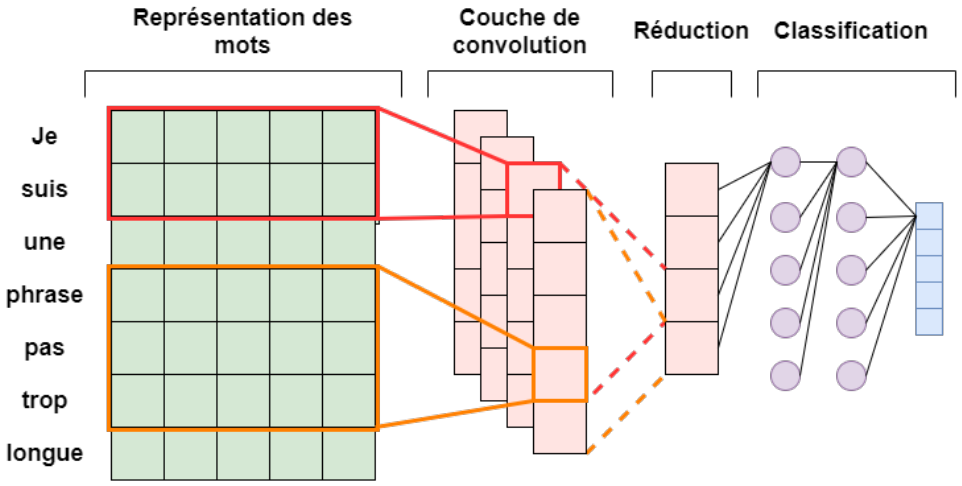

FIGURE 2.9 – Architecture de [38]

Une autre proposition [85] introduit aussi un réseau convolutif mais orienté sur un autre niveau de caractéristique. À la place d'un encodage des mots, celui-ci met en avant l'encodage des caractères sur un alphabet de taille m adapté au langage considéré. Cet alphabet est étendu pour inclure des caractères spéciaux et l'encodage se fait sous la forme d'un "one-hot vector" (comme pour la classification). La séquence de caractères devient

donc une matrice de taille fixe contenant l vecteurs (m). Le réseau lui-même se compose de six blocs d'une couche de convolution suivie d'une couche de réduction (Max-pooling). Il se termine par trois couches denses dédiées à la classification.

Il est également possible d'opérer une combinaison des réseaux récurrents et des réseaux convolutifs comme proposé avec une méthode de RCNN [44]. Celle-ci se compose de deux parties, une première permettant de calculer le contexte complet de chaque mot et la seconde dédiée à la classification. La première partie utilise un réseau bidirectionnel pour calculer le contexte droit et le contexte gauche de chaque mot du début jusqu'à la fin du texte. Les mots sont pour cela encapsulés avec la méthode Skip-gram de Word2Vec définie précédemment. Cette partie sert à renforcer la représentation des mots avec son contexte au sein du texte, qui devient alors la concaténation du contexte droit, de l'encapsulation du mots et du contexte gauche. L'ensemble des vecteurs concaténés remplace la couche de convolution traditionnelle pour enchaîner avec la couche de réduction qui a ici pour but de dégager les mots les plus importants du texte. Le réseau se termine ensuite avec une couche dense permettant le calcul du vecteur de sortie. L'architecture est résumée par la Figure 2.10.

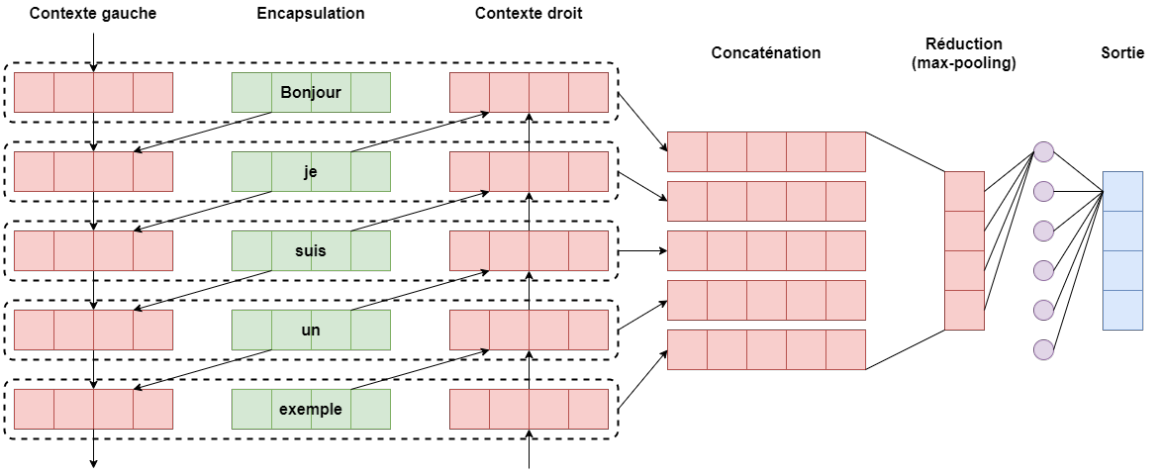

FIGURE 2.10 – Architecture de RCNN [44]

Ces méthodes de classification de texte ont déjà été appliquées avec succès dans un contexte d'application industriel comme c'est le cas pour CloudScan [60]. Il s'agit d'un système d'analyse de facture ne nécessitant aucune configuration ou pré-annotation. Il fonctionne par apprentissage d'un seul modèle global de facture qui est entraîné à partir des retours fournis par les utilisateurs finaux. Le modèle se base sur un réseau neuronal récurrent entraîné à extraire l'information de huit champs importants d'une facture non structurée (pas besoin d'une structure particulière de facture) au format PDF. Le retour est structuré au format XML, avec les champs remplis par les valeurs extraites.

Le processus est en six étapes :

— L'extraction du texte, réalisée soit en récupérant directement le texte du PDF soit via un OCR si l'option précédente est impossible.
— La génération de N-grams à partir des mots d'une même ligne. La longueur des N-grams est de quatre mots.
— Le calcul des caractéristiques pour chaque N-grams. Il y a trois catégories : tex-

tuelles, numériques et booléen. Ce calcul utilise une base de données contenant des informations générales (ville, pays, zip codes...).
— La classification de chaque N-gram parmi trente-deux champs d'intérêts ou à défaut dans le champ non identifié (soit trente-trois classes en tout)
— Un post-processus émet la décision finale pour chacun des champs parmi les N-grams candidats, en filtrant les candidats qui ne remplissent pas toutes les conditions du champ. Ces conditions sont vérifiées par des expressions régulières (cf : système à base de règles) ou, pour les champs sans liens sémantiques avec les autres, les assignations se font avec l'algorithme de Kuhn-Munkres (méthode Hongroise) [43]. Les expressions régulières sont renforcées pour gérer les erreurs OCR simples.
— La reconstruction du document avec les informations extraites.

Les différentes méthodes décrites dans cette section sont conçues pour être efficaces pour traiter des données équilibrées et avec beaucoup d'exemples pour les entraîner. Elles sont donc efficaces sur les classes fortement représentées dans le flux. De plus, les documents administratifs et industriels sont très majoritairement des documents avec beaucoup de texte (ou au moins très signifiant). Texte qui contient d'ailleurs les informations les plus importantes pour les distinguer les uns des autres. Dans ces conditions, le texte constitue la meilleure modalité pour différencier les classes du flux entre elles.

Cependant, il y a plusieurs problèmes majeurs à ces méthodes. Le premier est l'extraction du texte opérée par les OCRs. S'ils se sont beaucoup améliorés ces dernières années, ils restent assez sensibles aux bruits, aux dégradations et à la taille des caractères (peu être trop petite pour une extraction correcte). Sur ce point, la structure des documents est globalement plus résistante que le texte aux aléas d'une mauvaise dématérialisation. De plus, les OCRs utilisés par l'industrie ne peuvent pas encore reconnaître d'écritures manuscrites (cela nécessite des méthodes spécifiques). Autre problème, les documents peu verbeux, qui sont certes minoritaires, sont difficilement classifiables en utilisant uniquement le texte, qui en est presque absent. Enfin, ces méthodes ne répondent pas aux problématiques posées par le déséquilibre et l'incomplétude décrites dans la section 1.2.

2.1.3 Classification de documents administratifs à partir de l'image

L'état de l'art de la classification et du traitement de l'image offre un panel de nombreuses variations de réseaux de type convolutif, majoritairement en apprentissage profond [28, 86, 32, 70, 27]. Ces méthodes peuvent utiliser directement comme entrée la matrice formée par les pixels bruts de l'image, contrairement au réseau textuel. Les images sont standardisées au niveau de la taille et de la couleur, car les réseaux convolutifs exigent comme entrée une matrice de forme fixe (par exemple : 224x224x3 pour VGG16, soit une image en 224x224 pixel et une profondeur de couleur de 3, Rouge-Bleu-Vert). Les tailles sont souvent assez petites pour réduire le coût de calcul. Pour la couleur, si elle n'est pas considérée spécifiquement comme importante dans le contexte, celle-ci est ramenée à des niveaux de gris (toujours pour la même raison).

VGG16 [70] est un bon exemple d'architecture CNN profond, dont plusieurs éléments se retrouvent dans la plupart des autres réseaux convolutifs profonds de l'état de l'art. La figure 2.11 présente l'architecture globale du réseau VGG16 couche par couche. Dans un premier temps, ce schéma permet de remarquer que le réseau est organisé en cinq blocs de convolutions suivis d'une succession de trois couches denses en guise de classifieur. Chaque bloc contient deux ou trois couches de convolutions suivies d'une couche de réduction (dites de "max-pooling"). Cette couche de réduction diminue la taille des matrices de caractéristiques résultant du bloc en ne conservant que la valeur la plus élevée parmi un groupe de valeurs voisines (la taille est définie en 2x2 pour VGG16, ce qui divise par deux la taille de la matrice).

Avant les couches denses, il y a une opération d'aplatissement (flatten) qui consiste en une réduction de la matrice d'entrée à un vecteur unidimensionnel par un "aplatissement" des valeurs (ex : une matrice 3x3 devient un vecteur de longueur 9). L'objectif de cette opération est simplement de faire correspondre la sortie des couches convolutives avec l'entrée des couches denses. D'autres versions de cette architecture remplacent l'aplatissement par un global average-pooling, avec toujours pour but de transformer la matrice en vecteur mais cette fois en résumant l'information. Son fonctionnement est similaire au max-pooling, cependant la sélection de la valeur la plus élevée est remplacée par une moyenne. VGG16 montre également une haute précision autant sur la classification d'images que sur celle de documents [3].

Beaucoup de méthodes de réseaux convolutifs profitent des vastes corpus du domaine de l'analyse d'image comme ImageNet [64]. Ils permettent de pré-entraîner les poids utilisés comme initialisation pour de nouvelles tâches où le nombre d'exemples disponibles est plus faible. Cet usage du transfert de connaissances a pleinement montré son efficacité en accélérant la convergence des réseaux, en renforçant et en stabilisant les performances finales obtenues après l'entraînement. Les effets bénéfiques de cette pratique peuvent se voir à travers ResNet [41] sur la classification d'images. Ils sont également visibles sur les documents de la base RVL-CDIP avec la méthode de CNN holistic [27] qui utilise une initialisation des poids renforcée par un pré-entraînement sur la base ImageNet.

Cette méthode se base sur un système composé de cinq CNN combinés, quatre liés à une région particulière du document et le dernier s'opère sur l'image au complet. Les quatre régions proposées sont l'entête, le corps gauche, le corps droit et le pied de page. Les CNNs suivent l'architecture de la Figure 2.12 et se combinent par la concaténation

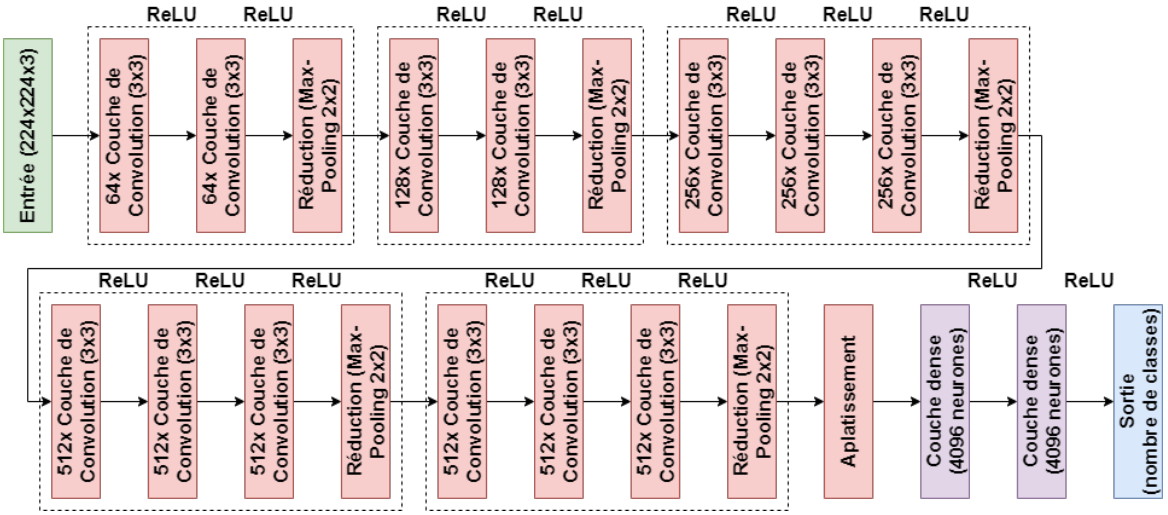

FIGURE 2.11 – Architecture d'un réseau VGG16 [70]

successive des vecteurs de caractéristiques extraites par chacun des cinq CNNs. Avant la combinaison, les vecteurs sont préalablement réduits par une ACP [80]. L'objectif de cette méthode est de classer le document en guidant l'extraction des caractéristiques par le découpage en région. Ce guidage est conçu pour prendre en compte l'aspect plus structurel des documents, avec les relations spatiales entre les régions, qui ne se retrouve pas forcément dans les images naturelles.

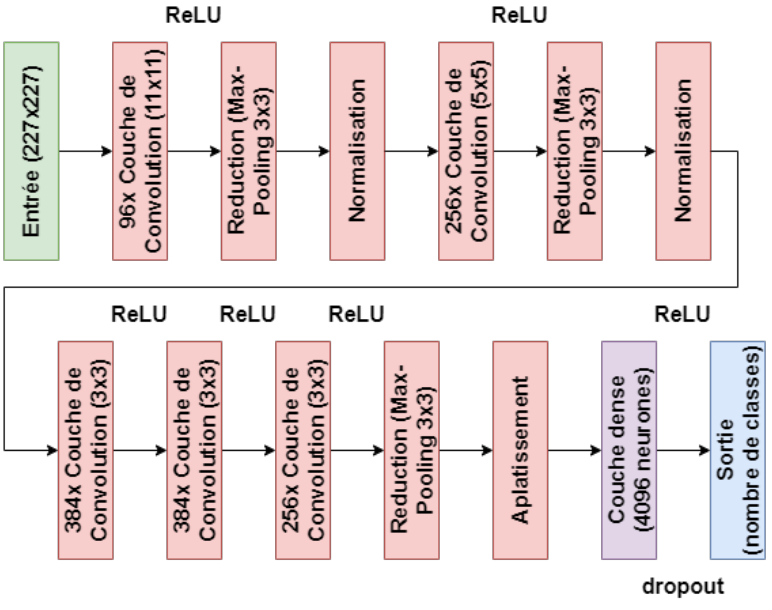

FIGURE 2.12 – Architecture d'un des réseaux convolutifs de HCNN [27]

Pour renforcer encore les performances des réseaux, il est également courant d'appliquer une réduction progressive de taux d'apprentissage. Elle peut s'effectuer en trois coups fixés à un certain nombre d'itérations ou d'étapes de la boucle d'entraînement. Une distribution typique : 80 itérations d'entraînement prévues avec un taux d'apprentissage

de 10^{-3}, puis aux itérations 20, 40 et 60 le taux d'apprentissage est divisé par 10 (donc au final 10^{-6}). Il y a par exemple, une méthode de ce type [3] conseillée pour les réseaux profonds convolutif en appliquant une réduction à chaque itération suivant l'équation 2.1.

$$nouveau\_taux = taux\_initial \times (1 - \frac{iteration\_courante}{iteration\_maximal}) \tag{2.1}$$

À l'instar de celles textuelles, ces méthodes sont efficaces pour les traitements des grandes classes du flux, même si à première vue elles semblent moins adaptées. Les documents administratifs et industriels ont fréquemment une mise en page et/ou une structure particulière qui devrait permettre de les distinguer visuellement. Cependant ce n'est pas toujours le cas, surtout que la réduction de définition de l'image et la standardisation de sa taille n'aident pas à les conserver. Quoi qu'il en soit, leur efficacité sur RVL-CDIP [3] montre qu'ils sont des solutions potentielles, preuve s'il en est de la résilience de ces méthodes. Le problème principal de ces méthodes reste là encore l'adaptation aux conditions des flux de documents, identique à celui des méthodes textuelles.

2.1.4 Classification multimodale de documents

En plus des approches déjà introduites, l'état de l'art présente également de nouvelles méthodes de réseaux neuronaux proposant une approche de classification combinant texte et image. Ces méthodes sont dites multimodales car elles exploitent deux modalités différentes. Cette dualité a pour objectif de compenser les faiblesses respectives des approches séparées, permettant de classer des types de documents plus variés qu'une méthode monomodale. Les caractéristiques des deux modalités peuvent être complémentaires dans certains cas offrant la possibilité d'utiliser des critères visuels pour différencier des classes floues du point de vue du texte et inversement (voir les exemples de cas difficiles de la section 1.2 (figures 1.2 et 1.3)).

Les solutions actuelles sont majoritairement des doubles réseaux [7, 84, 18, 5] (un pour l'image et un pour le texte) qui extraient chacun leurs caractéristiques indépendamment puis se fusionnent dans un méta-classifieur de couches denses, soit une fusion tardive. C'est le cas de la méthode NasNet+BERT [7] qui suit l'architecture de la Figure 2.13. Le classifieur image est un réseau convolutif inspiré de NasNet [86] et le classifieur textuel est le réseau de BERT [19] (décrit dans la section 2.1.2). La fusion s'opère par une moyenne entre les vecteurs de sortie des deux réseaux qui est ensuite envoyée en tant qu'entrée du méta-classifieur. Il est également possible de fusionner les vecteurs de sorties par une concaténation [18].

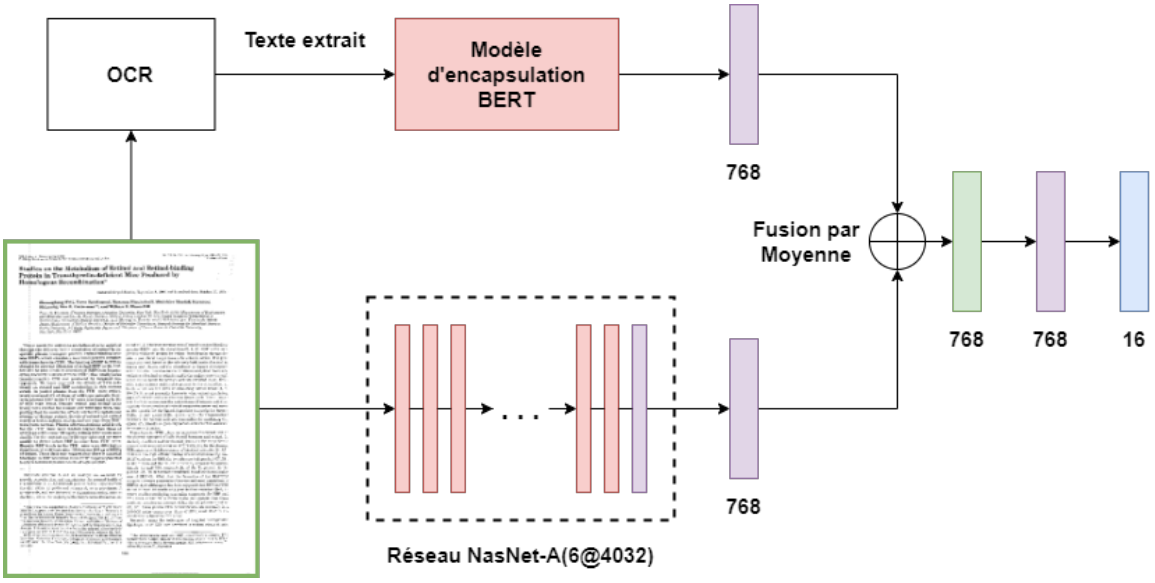

FIGURE 2.13 – Architecture multimodale NasNet+BERT [7]

La combinaison en double réseau n'est cependant pas la seule option proposée par l'état de l'art. LayoutLM [83] introduit lui une architecture inspirée de BERT mais renforcée à plusieurs niveaux par des caractéristiques visuelles. Dans un premier temps, les vecteurs d'encapsulation de chaque mot sont adjoints à leurs emplacements (position en x,y) pour générer de nouveaux vecteurs, via un réseau bidirectionnel proche de celui de BERT. Dans un second temps, un réseau de RCNN rapide [62] calcule une encapsulation de l'image correspondant au texte, à partir des régions d'intérêts.

Au regard des résultats, la combinaison des modalités semble être pertinente même si dans la plupart des propositions les modalités restent traitées séparément. En substance, les réseaux doubles n'extraient pas de caractéristiques multimodales à proprement parler mais un assemblage de caractéristiques de deux modalités différentes. Dans l'ensemble, ces méthodes montrent toutes de meilleures performances que leurs équivalents mono-modaux sur la classification de documents. Elles constituent donc d'excellents candidats pour la classification des grandes classes, mais restent tout autant dépourvues face au déséquilibre et à l'incomplétude du flux.

2.1.5 Modèle d'attention

Les modèles d'attention sont une évolution récente dans le champ des réseaux de neurones offrant une option pour résoudre l'un des plus gros points faibles de ceux-ci : leur effet boîte noire. Comprendre ce qu'utilise un réseau neuronal pour prendre une décision est difficile, étant donné que ces caractéristiques sont apprises statiquement par une boîte opaque qui n'offre pas vraiment de retour sur ce qui se passe à l'intérieur, si ce n'est les résultats renvoyés par les différentes couches. Les mécanismes d'attention semblent pouvoir contrebalancer en partie cet effet tout en améliorant la qualité des caractéristiques utilisés par le réseau dans son processus de décision [33].

Bien que ce mécanisme soit maintenant utilisé pour résoudre une grande variété de problèmes comme la génération de description d'image ou l'encapsulation de mots (BERT), il a initialement été conçu pour de la traduction de texte. Pour cela, il est utilisé avec un réseau de type encodeur-décodeur [15]. Il s'agit d'un double réseau, dont le premier sert à encoder une séquence (à traduire et dite source) en une représentation vectorielle et le second utilise ce vecteur pour générer une nouvelle séquence (la traduction, dite cible). Cependant, ce type de modèle n'était à l'origine fiable que sur de petites séquences dû à la compression des informations de la séquence en un vecteur de taille fixe. Pour permettre au modèle de mieux apprendre les relations entre la source et la cible, la solution proposée a été d'utiliser les sorties des couches cachées de l'encodeur. Elles prennent alors la forme d'une séquence d'annotations [6]. L'idée est de conserver ces informations à travers un vecteur de contexte calculé par un réseau dense et une fonction d'activation softmax (qui a donc comme entrée la séquence d'annotation issue des couches cachées de l'encodeur). Le vecteur de contexte est ensuite ajouté à l'entrée du décodeur en les concaténant. Ce vecteur de contexte permet d'évaluer à quel point les éléments de la séquence source correspondent à la séquence renvoyée par le décodeur. Bert [19] se sert d'un système d'attention similaire, mais les multiplient pour les utiliser en parallèle suivant la Figure 2.14. Cette organisation a pour but de renforcer l'effet du mécanisme.

Les systèmes d'attention sont également utilisés pour le traitement d'image avec un fonctionnement proche [23, 33]. Ils se déploient alors à partir des couches de convolution du réseau pour renforcer la dernière couche dense, comme l'illustre la Figure 2.15. L'idée est d'appliquer le mécanisme d'attention à plusieurs niveaux de résolutions du réseau. Ces niveaux de résolutions correspondent aux sorties des trois derniers blocs de convolutions (ceux qui extraient les caractéristiques de haut niveau et donc les plus informatives). Grâce à cela, la décision finale peut être renforcée par des caractéristiques importantes plus locales de l'image mises en saillance par le système d'attention [23]. Ce type de système permet de voir ce que le réseau considère comme une partie importante de l'image, en analysant le masque d'attention généré par les différents modules (cet effet se retrouve également pour le texte au niveau des mots mis en avant par l'attention). Il permet également de concentrer la décision du réseau sur les parties les plus importantes de l'image, écartant ainsi les informations hors-sujet ou portant à confusion [33].

Les propriétés des modèles d'attention ont beaucoup d'intérêt dans notre contexte. Les documents étant des images et des textes où seule une petite partie est véritablement décisive pour l'identifier. Dans les faits, une part importante du contenu d'un document

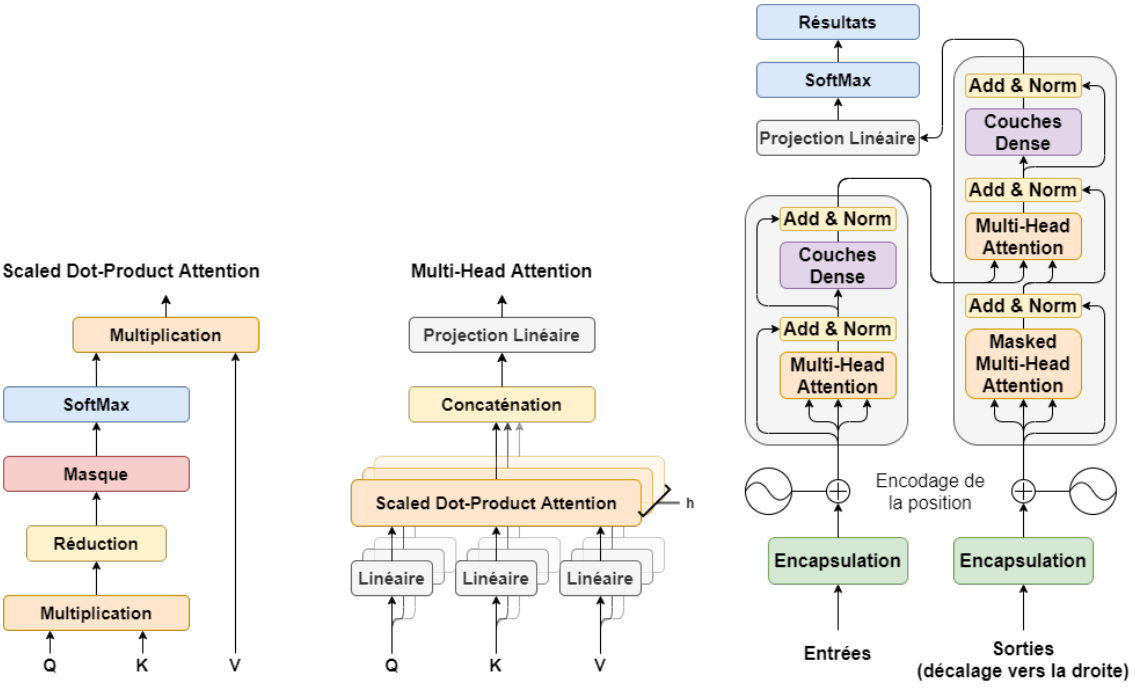

FIGURE 2.14 – Système d'attention et architecture de l'encodeur utilisé par BERT [77]

est trop spécifique (comme le détail des articles d'une facture, d'un devis ou d'un bon de livraison, les closes juridiques d'un contrat). Si ces informations sont essentielles à la compréhension du document lui-même, ils n'aident pas forcément à en identifier la nature, voire provoquent de la confusion avec les classes proches (comme pour le cas Facture-Bon de livraison de l'introduction (figure 1.2)). Focaliser l'attention du réseau sur les informations les plus discriminantes (comme le titre) semble donc très pertinent. D'autant plus qu'avec le déséquilibre le modèle ne pourra pas lisser la pertinence des caractéristiques qu'il extrait pour les classes minoritaires via des statistiques larges. Le contraindre avec ce type de système pourrait faciliter l'apprentissage de ces classes.

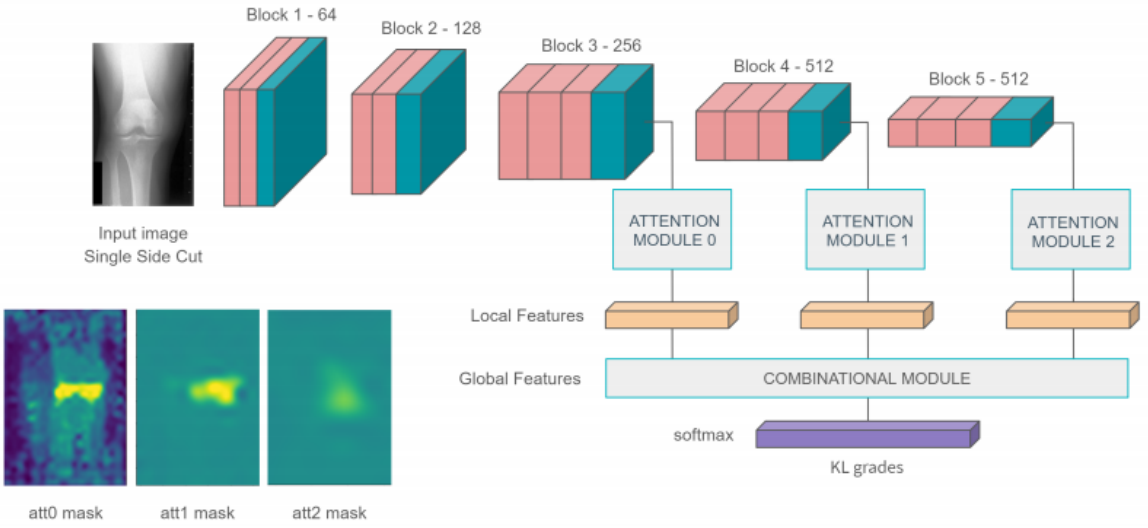

FIGURE 2.15 – Système d'attention pour VGG [23]

2.2 Classification déséquilibrée de documents

Pour rappel, la classification déséquilibrée est ici un contexte d'entraînement particulier pour les méthodes de classification par apprentissage, où il y a un nombre inégal d'échantillons d'entraînement par classe. C'est un contexte très présent dans le cadre industriel, où les échantillons disponibles proviennent des chaînes de productions. Dans la littérature, il n'y a pas beaucoup de publications autour de la classification déséquilibrée de documents avec des machines à apprentissage statistique. Néanmoins, il est possible de ressortir deux types de solutions à ce problème [35, 47]. Les premières consistent à renforcer l'entraînement des classes sous-représentées en pondérant le coût des erreurs. Les secondes cherchent plus à rééquilibrer le corpus par une augmentation des données à partir de celles disponibles et/ou de connaissances à priori.

2.2.1 Renforcement des classes sous-représentées

Ce premier type de méthode propose de compenser le déséquilibre dans l'entraînement des classes en renforçant le poids de celles sous-représentées dans le calcul du gradient. Pour rappel, il s'agit de la valeur de correction calculée par la fonction de coût et appliquée sur les paramètres du réseau. Le problème que pose la sous-représentation pour les réseaux étant que ces classes n'influencent pas assez la convergence des paramètres et des poids des neurones, ce renforcement est là pour le compenser directement. Pour renforcer le poids des classes sous-représentées, l'option la plus simple est de pondérer la fonction de coût en augmentant le gradient dans les cas d'erreur les impliquant.

Il est également possible de le faire avec de l'apprentissage renforcé [49]. La problématique est définie alors comme un jeu de devinettes qui se décompose en une séquence de décision, chaque décision correspondant à la classe d'un échantillon. Ensuite, la récompense varie entre positif et négatif selon si la décision est la bonne ou non. Dans le cas où la vraie classe fait partie de celles sous-représentées, alors la récompense est 1 ou -1.

Dans le cas contraire, un facteur de réduction est appliqué à la récompense. Plus la classe est représentée dans le corpus, moins la récompense est élevée, le nombre d'échantillons de la classe compensant la réduction. L'objectif final est d'obtenir le score final le plus élevé possible. Ce processus de décision s'applique sur un réseau neuronal en remplaçant la traditionnelle fonction softmax de la dernière couche dense.

2.2.2 Augmentation des données

L'autre option proposée dans l'état de l'art correspond aux méthodes permettant l'augmentation des données. Elles peuvent être utilisées pour compenser la faible représentation d'une partie des classes par la génération de nouveaux échantillons à partir de ceux disponibles. À noter que pour rééquilibrer, il est également possible de réduire le nombre d'exemples des classes sur-représentées. Par contre, cette méthode pose la problématique de la représentativité statistique d'un échantillon pour sa classe (soit les informations qu'il apporte sur la classe) [51].

L'état de l'art présente plusieurs méthodes applicables aux documents [48]. Parmi elles, se trouvent des méthodes de génération par transformation. Les versions les plus classiques proposent de simples transformations géométriques appliquées sur l'image (typiquement des rotations ou des décalages). Les autres versions introduisent des déformations ou des bruits pouvant simuler des altérations et des dégâts subis par l'image, cependant celles-ci doivent rester mesurées pour ne pas trop dépasser le cadre de la classe.

Il existe également un type de réseau neuronaux spécialement conçu pour la génération d'images : les "Generative Adversarial Networks" ou GAN [50]. Ces réseaux se divisent en deux parties : la première chargée de générer des exemples, la seconde de discriminer les échantillons générés des vrais. L'entraînement est donc fait par opposition entre les deux parties qui se renforcent mutuellement en se confrontant. Bien qu'elles soient très performantes pour générer des images, ces méthodes statistiques peuvent difficilement être fiables avec aussi peu d'exemples pour les entraîner. En plus, les documents sont des images particulièrement complexes à modéliser, ce qui n'aide pas. Les GAN ne semblent pas adaptés pour augmenter les classes sous-représentées.

Il y a également un autre problème avec la génération d'exemples. Les classes les moins représentées ont un nombre d'échantillons très faible (un ou deux exemples) et finiront donc par compter beaucoup plus de documents générés que de documents réels. Cette sur-génération risque "d'enfermer" la classe sur les quelques exemples qui ne sont pas forcément représentatifs de sa diversité interne. En conséquence de quoi, le réseau pourrait ne pas généraliser la classe aux documents proches, mais se limiter à ceux quasiment identiques. Si cet effet est un problème mineur pour les classes dont la variété interne est très faible (car très structurées comme les cartes d'identité), ce n'est pas le cas pour les classes avec une variété plus importante (facture, lettre ...).

Ces deux types de solutions, et les méthodes qui les accompagnent, sont de bons candidats pour répondre au déséquilibre. Plusieurs versions d'entre elles semblent pouvoir facilement se combiner avec n'importe quelles autres méthodes. Par contre, elles n'apportent rien contre l'incomplétude et ne semblent pas non plus offrir d'option d'évolution allant dans ce sens.

2.3 Apprentissage avec peu d'exemples

Dans les sections précédentes nous avons vu des solutions possibles pour résoudre les problèmes de la classification des grandes classes et du déséquilibre. Il reste cependant le cas des très petites classes et celui de l'incomplétude. Plusieurs propositions existent dans l'état de l'art, tentant d'apporter des solutions à ces problématiques. Des méthodes essayent d'apprendre autant que possible du strict minimum d'exemples et d'autres essayent d'inférer l'apprentissage des classes manquantes. Dans les prochaines sections, nous commencerons par l'analyse du second type de méthodes.

2.3.1 Zero-shot learning

Le "zero-shot learning" est un défi issu du traitement de l'image [82]. Il consiste à effectuer des prédictions d'une partie des classes sans apprentissage d'exemples étiquetés, soit de la classification avec modèle entraîné sur une base d'apprentissage qui ne contient pas toutes les classes du corpus de test.

La stratégie principale utilisée pour ce défi est d'utiliser deux sources d'information différentes et de compenser l'incomplétude de la première par la seconde. Par exemple, pour classer une image de panda sans y avoir eu accès, le modèle s'appuie sur une description textuelle de ce qu'est un panda pour le reconnaître. La problématique est alors transformée en une autre : le transfert de connaissances depuis un espace de représentation sémantique des images (complet) vers un espace de représentation visuel des images (incomplets). L'espace de représentation sémantique peut être une liste d'attributs, une description, des annotations, etc.

L'une des méthodes les plus présentes dans l'état de l'art pour opérer l'échange de connaissances entre les deux domaines est l'entraînement d'une fonction de transfert. ALE [4] est un bon exemple de ce type de méthode. Cette solution se concentre sur la création d'un espace de représentation de labels issus d'un ensemble d'attributs liés aux images à classer. De l'autre côté, des caractéristiques sont extraites de l'image permettant l'établissement de deux espaces de représentation comme l'illustre la figure 2.16. Ainsi, l'image est définie d'un côté par ses caractéristiques et de l'autre le label est défini par ses attributs. À partir de cela, la solution consiste à entraîner une fonction de transfert permettant de relier les caractéristiques aux attributs et donc par extension de relier les images aux labels.

DeViSE [21] se base sur un principe similaire, un réseau convolutif profond de classification d'image (cf Section 2.1.3) pré-entraîné sur ImageNet [64] permet l'extraction des caractéristiques visuelles et un modèle de langue Skip-gram de Word2Vec (cf Section 2.1.2) entraîné sur Wikipédia pour la partie textuelle. Une transformation linéaire est appliquée sur l'encapsulation de l'image générée par le CNN pour que les dimensions du CNN et du modèle de langue correspondent. Enfin, la liaison se fait par une mesure de similarité entre la sortie de chacun des réseaux. L'ensemble est alors réentraîné avec une fonction de coût permettant de renforcer la similarité par produit scalaire entre les deux sorties : image et label. Dans les deux cas, le zero-shot learning est obtenu par la généralisation des liens entre images et labels offrant ainsi la possibilité d'associer des couples label-image qui n'ont pas été entraînées.

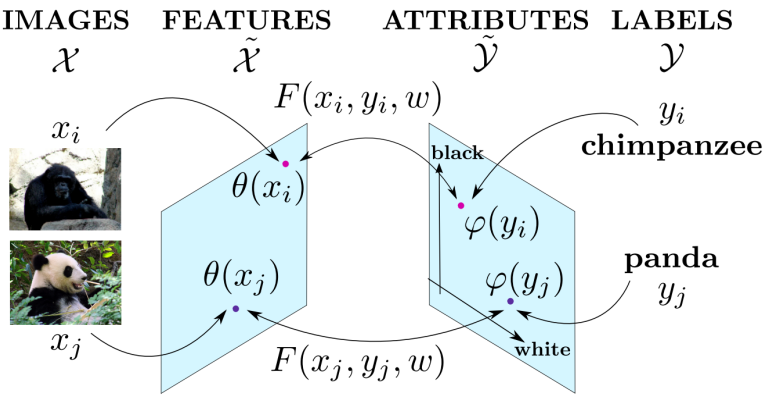

FIGURE 2.16 – Illustration de la stratégie de ALE [4]

SYNC [12] propose une méthode différente qui cherche à "aligner" les espaces sémantiques et visuels (voir Figure 2.17). Le premier peut être généré de la même manière que les deux méthodes précédentes (soit avec des attributs, soit avec un modèle de langue comme Word2Vec). Le second peut être généré par n'importe quel extracteur de caractéristiques visuelles, mais les réseaux convolutifs profonds semblent être ceux qui offrent les meilleurs résultats. Le système ajoute aux deux espaces un groupe de classes dites "fantômes" (*b* et *v*) qui ne correspondent pas à de véritables classes mais forment avec les classes "réels" (*a* et *w*) un graphe bipartite pondéré utilisé pour alignement. L'objectif est alors d'apprendre l'alignement entre les deux graphes tout en minimisant la distorsion et ainsi relier les classifieurs de l'espace visuel (model space) et les classes de l'espace sémantique.

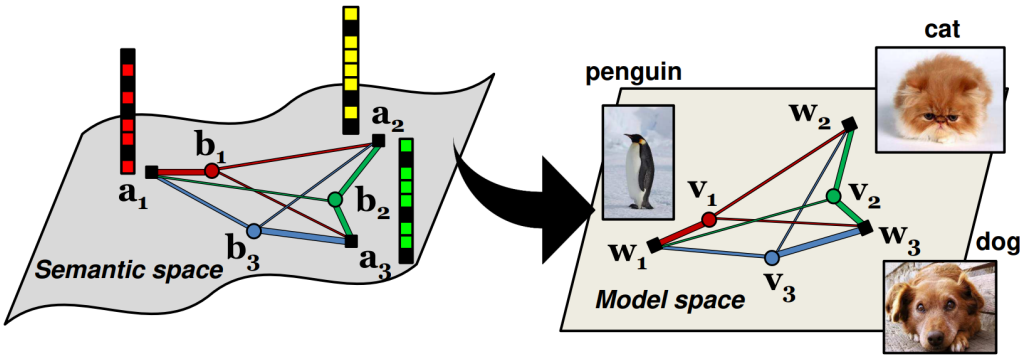

FIGURE 2.17 – Illustration du principe de fonctionnement de SYNC [12]

L'adaptation de ce défi à notre problématique paraissait dans un premier temps adéquate, surtout pour le problème de l'incomplétude. Cependant, les méthodes développées pour répondre aux contraintes se basent toutes sur l'existence d'informations tierces sur les classes non présentes. Malheureusement dans notre contexte, où il est considéré comme impossible de disposer d'informations à priori sur ces classes, il est impossible donc de remplir cette condition. En conséquence, ces méthodes sont inadaptées à notre contexte tant que celles-ci nécessiteront une source d'information tierce. Reste à voir si,

dans le futur, de nouvelles méthodes n'incluant pas ce défaut apparaîtront dans ce domaine en constante évolution.

2.3.2 One/Few-shot learning

Le zero-shot étant hors-jeu, il reste les méthodes d'apprentissage sur un minimum d'exemple avec notamment le "one-shot learning". Il s'agit d'un défi issu du traitement de l'image. Il consiste à entraîner un modèle même avec un seul exemple pour certaines classes. Le "few-shot learning" suit le même principe mais en permettant un peu plus d'exemples par classes (autour d'une dizaine au maximum). La quantité la plus utilisée dans l'état de l'art, hors "one-shot", est le "five-shot", soit un entraînement avec cinq exemples. Toute la difficulté du défi se concentre alors sur l'apprentissage du maximum d'informations à partir des quelques exemples disponibles, soit la modélisation la plus complète possible à partir du minimum de données.

Les stratégies utilisées pour tenter de résoudre ce défi sont nombreuses et très diversifiées, mais deux axes principaux se dégagent de l'ensemble. Le premier se concentre sur une approche par inférence Bayésienne et du méta apprentissage. Le second tente une adaptation de méthodes d'apprentissage comme les réseaux neuronaux en centrant l'entraînement sur des calculs de différences ou en les complétant avec d'autres approches les renforçant. Les méthodes ne se privent pas de mélanger les deux approches, mais en se basant plus sur l'une que l'autre.

Le principe de l'inférence bayésienne est d'utiliser le théorème de Bayes pour calculer la probabilité d'une hypothèse (l'appartenance à une classe dans notre cas) sachant des paramètres, des caractéristiques ou des éléments précédemment rencontrés, etc. L'intérêt de ce type d'inférence est qu'elle ne nécessite pas beaucoup d'exemples pour fonctionner et qu'elle peut se renforcer avec le temps, d'où son utilisation dans ce contexte. L'inférence bayésienne est utilisée pour la catégorisation d'objet dans une image en "one-shot learning" [20], où la probabilité d'appartenance à une classe est comparée avec celle de faire partie de l'arrière-plan de l'image. L'image est divisée en région susceptible d'être un objet par un détecteur d'éléments saillants. Les caractéristiques utilisées pour l'inférence sont la position et l'apparence de ces régions. La modélisation est faite en utilisant un graphe en étoile.

L'inférence bayésienne est également utilisée pour la classification de caractères manuscrits en one-shot [45]. Cette méthode, appelée BPL (pour Bayesian Program Learning), se base sur l'apprentissage d'un programme stochastique simple permettant la représentation de concept sous la forme d'un modèle de génération (ex : un modèle de génération de "A"). Pour conceptualiser un caractère manuscrit, celui-ci est fragmenté en morceaux sur trois profondeurs selon son tracé (Voir Figure 2.18) : les relations (IV), les parties (III) et les sous-parties (II). Les sous-parties d'un type correspondent à des tracés primitifs (I) communs entre les différents types de caractères. Le schéma obtenu par ce découpage permet de définir une procédure de génération d'exemples de ce caractère. La procédure inclut des transformations aléatoires pour créer de la diversité dans les exemples générés. Pour permettre l'inférence, le meilleur générateur est sélectionné parmi l'ensemble des modèles qui auraient pu générer le caractère analysé. La classification se fait en comparant trois images générées à partir du modèle de génération et l'image à classer. La comparaison se fait par un score de probabilité (log posterior predictive probability). Cette méthode semble très difficile à mettre en place pour de la classification de documents bien plus complexes que de simples caractères.

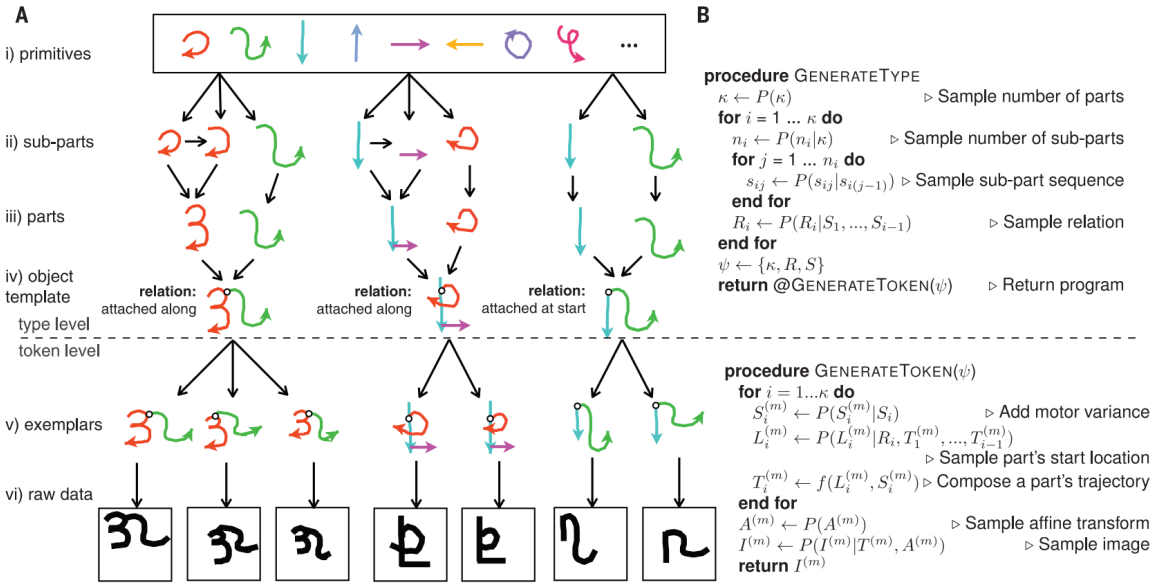

FIGURE 2.18 – Modèle de génération de caractères manuscrit de BPL. (A) Illustration du principe des modèles de génération avec l'assemblage de traits "primitifs". (B) Pseudo-code permettant la génération de nouveaux types de caractères (GenerateType) et de nouveaux exemples d'un type (GenerateToken) [45]

Plusieurs approches de l'état de l'art proposent d'adapter les réseaux neuronaux au problème de l'apprentissage avec peu d'exemples par des modifications structurelles comme les Neural Turing Machine [66], les réseaux siamois [39] et les réseaux prototypiques [71]. Ces méthodes sont évaluées principalement sur la base de données Omniglot composée de caractères manuscrits.

Les Neural Turing Machine (NTM) [25] sont des réseaux neuronaux récurrents renforcés par l'introduction d'un système de mémoire inspirée des Memory Augmented Neural Network [79]. Les NTM se composent en premier lieu d'un contrôleur qui n'est simplement qu'un réseau neuronal classique (en général un LSTM, mais il n'a pas obligatoirement besoin d'être récurrent). L'autre composant est la mémoire externe, sa taille $M \times N$ est fixée sur un nombre d'emplacements N et chaque emplacement peut contenir un vecteur de données M (voir Figure 2.19). Le contrôleur interagit avec la mémoire par un ensemble de têtes reliées à son entrée et à sa sortie. Leur nombre est fixé en amont et chacune d'entre elles est dédiée à une action soit de lecture, soit d'écriture. L'opération de lecture consiste à calculer une combinaison convexe des vecteurs de la mémoire, pondérée par des poids entraînés spécifiques à la tête de lecture. L'opération d'écriture s'opère en deux temps. Elle commence par un "effacement" puis est suivie par un "ajout". Ces deux actions sont pondérées par la matrice de poids associée à la tête d'écriture. Le processus d'utilisation de la mémoire est guidé par un système d'adressage avec plusieurs options de fonctionnement possibles.

Une méthode utilisant les NTM propose de résoudre la problématique du one-shot learning par du méta-apprentissage sur la distribution au sein et entre des jeux de données. Pour cela, le réseau reçoit en entrée l'image à prédire et la prédiction précédente $(image_t; label_{t-1})$. La méthode utilise également un système d'adressage qui permet d'écrire soit sur l'emplacement le moins utilisé, soit sur le plus récemment utilisé. En d'autres termes, soit inscrire une nouvelle information en écrasant la plus ancienne, soit modifier

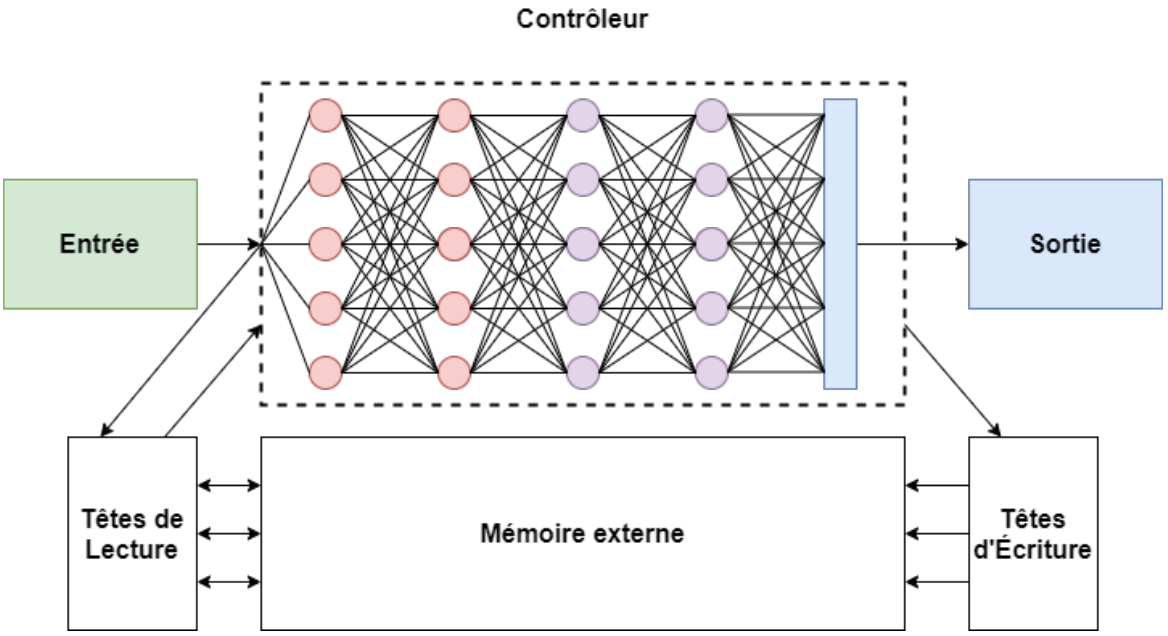

FIGURE 2.19 – Architecture d'une Neural Turing Machine [25]

celle qui a été renvoyée par la tête de lecture (et donc liée à la donnée courante). Le choix entre ces deux options est obtenu par interpolation entre les matrices de poids des têtes de lecture et d'écriture.

Les réseaux siamois [39] sont une autre option possible pour réaliser du one-shot learning. Il s'agit de deux réseaux distincts mais de structure similaire qui fusionnent leurs dernières couches pour renvoyer une sortie unique. Ces réseaux sont conçus pour calculer des scores de proximités entre deux entrées ce qui permet de réaliser du one-shot. L'architecture proposée se compose de deux réseaux convolutifs profonds identiques, suivant la figure 2.20, dont l'entrée est l'image à classer pour le premier réseau et un exemple de la classe considérée pour le second. Les réseaux sont eux-mêmes composés de quatre couches convolutives avec fonction d'activation ReLU entre coupées de réduction de type max-pooling. Les réseaux se terminent par des couches denses où les deux se rejoignent sur la dernière. Pour l'apprentissage, le réseau siamois utilise une fonction de coût et un optimisateur spécifique pour adapter l'entraînement au contexte.

La dernière architecture est celle des "réseaux prototypiques" (*Prototypical Network*, abrégé ici en ProtoNet) [71]. Cette méthode de few-shot learning se base sur un réseau convolutif profond entraîné pour générer une représentation multidimensionnelle des images d'entrées. Dans un premier temps, le réseau calcule le prototype de chaque classe, puis la classification s'opère par un calcul de la distance euclidienne entre les vecteurs de représentation des données à classer et les prototypes des classes. Une donnée est assignée à la classe du prototype le plus proche dans l'espace de représentation. Les prototypes des classes sont calculés à partir d'un petit ensemble d'exemples connus de la classe. Ils se présentent sous la forme de la moyenne entre les vecteurs de l'ensemble, soit le centroïde des exemples connus au sein de l'espace de représentation du réseau convolutif. Le processus de classification est résumé par la Figure 2.21. Pour s'entraîner, le système prend une partie de l'ensemble d'entraînement qu'il divise aléatoirement en deux groupes. Le premier sert à calculer le prototype de chaque classe et les autres de requêtes. Le coût

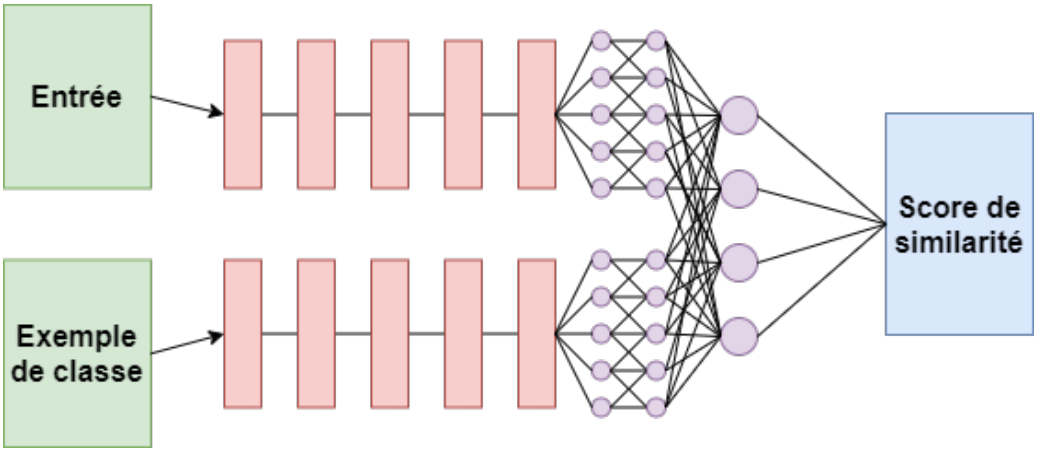

FIGURE 2.20 – Architecture d'un réseau siamois [39]

est calculé à partir des requêtes puis utilisé pour optimiser les paramètres. Le processus se répète avec un nouveau groupe et une nouvelle répartition aléatoire à chaque itération. Cet entraînement a pour but d'optimiser l'écart entre les vecteurs selon leurs classes dans l'espace de représentation, en rapprochant les vecteurs d'une même classe. Cette méthode a également été formalisée pour du zéro-shot learning [71] en générant les prototypes de classes à partir de métadonnées.

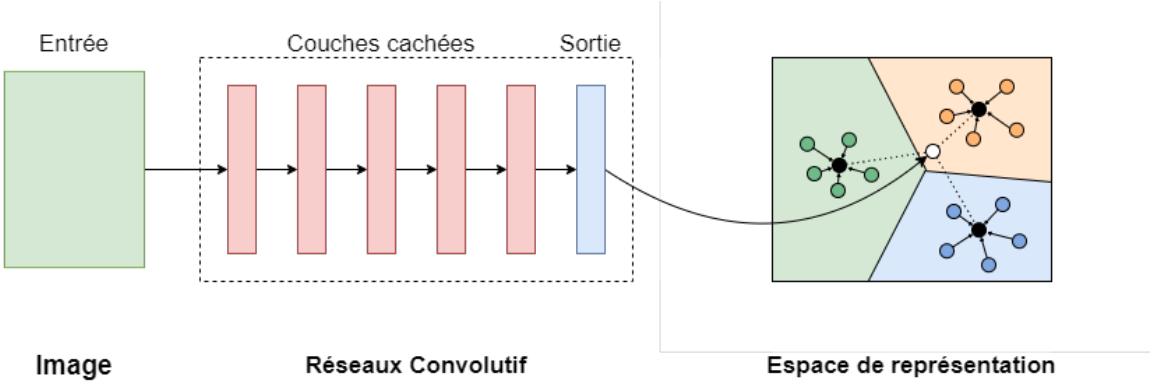

FIGURE 2.21 – Processus de classification avec ProtoNet [71]. Dans l'espace de représentation, les points noirs sont des prototypes de classes, les points colorés sont les exemples connus et le point blanc est la donnée à classer.

En conclusion, il y a une grande variété de méthodes de "one-shot learning", toutes conçues pour permettre l'entraînement d'une classe à partir d'un seul ou d'un petit groupe d'exemples. Cette faculté est très utile dans le contexte des flux de documents où une part non négligeable des classes correspond à cette description (voir section 1.2). Néanmoins, ces méthodes affichent des performances globalement inférieures pour la classification du groupe des classes les plus représentées, qui constitue pour rappel la majorité du flux. Cette perte de performance se doit d'être supprimée ou au moins fortement réduite pour pouvoir utiliser ces méthodes dans notre contexte.

2.4 Apprentissage incrémental

L'apprentissage incrémental se déroule, comme son nom l'indique, par incrément (généralement l'arrivée d'un nouvel élément ou groupe d'éléments à analyser). Chaque incrément est fini par un apprentissage plus ou moins important du modèle (qui peut être négligeable). La particularité de ce type d'apprentissage est donc que la phase d'entraînement n'est pas véritablement séparée de la phase d'utilisation, le modèle se renforce dès qu'il en a l'occasion. Cette particularité pose de nouveaux questionnements notamment au niveau de la croissance du modèle (qui peut alors devenir infini). Se retrouvent dans cette catégorie de nombreuses adaptations d'algorithmes préexistants, comme les k moyen incrémental [1] ou les machines à vecteurs de support (SVM) incrémentaux [46] qui ne seront pas tous détaillés ici.

2.4.1 Neural gas

Les algorithmes de classification du type Neural Gas (NG) sont inspirés des modèles du système neuronal humain. L'idée est de représenter les classes désirées par des centroïdes (neurones). Le nombre de centroïdes est fixe et représente le nombre de classes à considérer. Chaque donnée rencontrée au cours de l'exécution est placée dans un espace de représentation composé des caractéristiques utilisées pour les décrire. Ces données sont alors classées en fonction de leur proximité avec les centroïdes. Le centroïde le plus proche dans l'espace donne la classe de la donnée observée.

Dans la version non incrémentale, les centroïdes sont appris au cours d'une phase d'apprentissage non supervisée. La position des centroïdes est adaptée à chaque itération de manière plus ou moins importante en fonction de sa proximité avec la donnée courante et de son appartenance, ou non, à la même classe. Si le centroïde appartient à la même classe que la donnée alors il se rapproche dans l'espace, dans le cas contraire il s'éloigne.

Dans la suite de ces algorithmes, un nouveau type a été développé, il est dit Growing Neural Gas (GNG). L'intérêt de cette nouvelle version vient du fait que le nombre de centroïdes n'est plus fixe. À l'initialisation, un nombre initial de centroïdes est donné mais ce nombre peut désormais être modifié avec le temps. Les neurones sont reliés entre eux par des arêtes pour former un graphe. Ces arêtes vieillissent à chaque itération et sont rajeunies à chaque utilisation. Si une arête est trop vieille elle est supprimée et si un neurone se trouve isolé du graphe, il est également éliminé. Ce système permet de limiter le nombre de neurones présents et ainsi d'alléger le programme.

Les versions incrémentales de ces algorithmes proposent l'introduction d'un rayon d'action fixe ou adaptatif selon les cas. Ce rayon limite la sphère d'influence de chaque centroïde. Ainsi dans le cas où une donnée se trouve hors de tout rayon d'action, celle-ci est considérée comme étant une nouvelle classe et provoque la création d'un nouveau centroïde.

AI2NG [11] est une méthode de classification de flux de documents directement inspirée des algorithmes d'Incremental Growing Neural Gas (IGNG). Elle utilise une approche par apprentissage actif séquentiel semi-supervisé [10].

L'apprentissage actif désigne une pratique consistant à prévoir dans l'algorithme une évaluation de l'importance d'un exemple durant son apprentissage. Si l'exemple est important, le système demande à un opérateur humain la vérité terrain de cet exemple. Dans

le cas d'un apprentissage actif, le corpus d'apprentissage est non-étiqueté (ou au moins partiellement), les étiquettes « importantes » étant demandées par l'algorithme. Deux options sont possibles pour le retour de l'opérateur : statique ou séquentiel. Le retour statique correspond à une réponse en une seule fois, là où le retour séquentiel s'effectue au fur et à mesure de l'apprentissage.

L'apprentissage semi-supervisé mélange dans son corpus d'apprentissage des éléments étiquetés et non étiquetés. L'objectif est de combiner, autant que possible, les forces des apprentissages supervisés et non-supervisés. L'apprentissage supervisé consiste à fournir au programme un corpus annoté contenant donc la vérité terrain. Cette méthode est « coûteuse », relativement à la taille du corpus annoté mais elle permet un apprentissage optimal, ce qui tend à augmenter la performance de l'algorithme. L'apprentissage non-supervisé utilise un corpus sans annotation. Cette méthode est peu coûteuse mais ne permet que difficilement l'orientation de l'apprentissage et sa compréhension. La performance est en général réduite par rapport à la version supervisée.

A2ING s'inscrit comme un algorithme du type IGNG, mais une version avec un entraînement adaptatif (noté AING pour Adaptive Incremental Neural Gas). Celui-ci s'initialise avec quelques données étiquetées correspondant à la première instance du flux. À chaque itération, l'indice d'incertitude de la nouvelle instance est calculée pour déterminer si le programme doit faire appel à l'opérateur ou conserve sa prédiction. La prédiction se fait par rapport à la classe du neurone le plus proche.

Pour déterminer si l'intervention de l'opérateur est nécessaire, il faut vérifier que la valeur ne se trouve pas également dans le rayon d'action d'un autre neurone (voir Figure 2.22). La détection de nouvelle classe se fait avec le rayon d'action des neurones à l'instar de GNG. Cependant, celui-ci est dynamique pour chaque neurone. Lorsqu'une valeur extérieure est étiquetée comme appartenant à la même classe que le centroïde, son rayon augmente. Si au contraire une valeur se trouve dans le rayon du neurone et se révèle ne pas appartenir à la même classe, alors celui-ci est réduit. Cette réduction/augmentation est relative à la distance qui la sépare du centroïde.

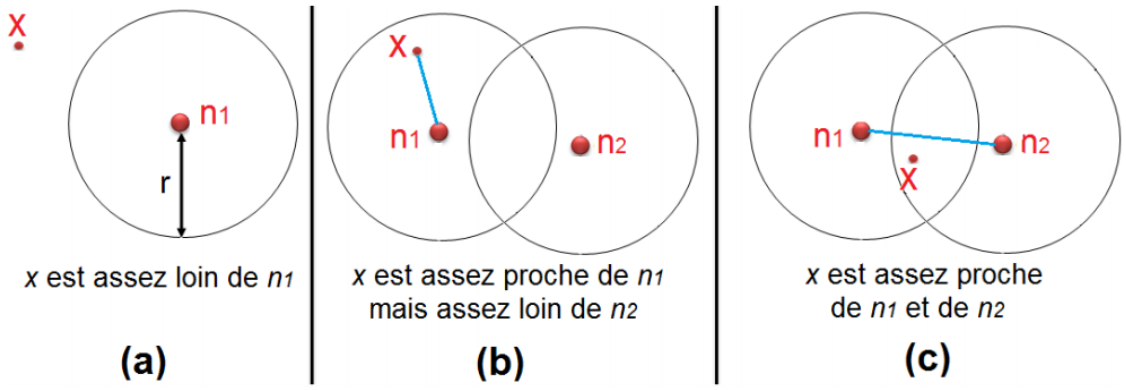

FIGURE 2.22 – Illustration des trois cas possibles lors de l'apparition d'une nouvelle donnée pour A2ING d'après [11]. Dans le cas (a) et (c), l'appel à l'opérateur est à considérer. Dans le cas (b), le système peut classer seul.

Pour contrebalancer les potentielles erreurs d'étiquetage effectuées par l'opérateur, A2ING propose de calculer le degré de désaccord entre la probabilité de la classe de l'étiquette retournée et la probabilité de la classe prévue.

A2ING est adapté aux conditions particulières des flux de documents : la forte hétérogénéité du flux, l'inégalité de représentation des classes, les faibles connaissances à priori et l'apparition subite de nouvelles classes. Cependant, il n'offre pas des performances satisfaisantes d'un point de vue industriel et ne dispose pas de gestion de son expansion dans le temps. L'article ne présente pas vraiment ni son extracteur ni les caractéristiques utilisées. Cette méthode exige également une intervention humaine et n'est donc pas pleinement automatique.

L'état de l'art montre que ce type de méthode a déjà été appliqué dans des contextes industriels, comme c'est le cas pour INDUS [17]. Il s'agit d'un système de classification et d'extraction de contenu à apprentissage semi-supervisé et incrémental pour gestionnaire numérique de courriers. Celui-ci ne demande que peu de configuration et est capable de s'adapter à de nouveaux éléments de manière quasi instantanée.

La classification des documents est assurée par A2ING renforcé par un système de souvenir couplé à une composante de vérification de qualité/renforcement. L'idée derrière ce système est de s'assurer autant que possible que chaque ajout au sein du modèle de A2ING ne provoque pas de régression des performances. Les "souvenirs" se constituent d'un ensemble d'exemples, qui sont les couples documents-classes les plus informatifs rencontrés selon l'algorithme de classification. La quantité d'exemples est limitée, si il n'y a plus de place, le nouvel exemple remplace un plus ancien de manière aléatoire. Les couples permettent, en les réinjectant dans l'algorithme, de vérifier que les classes retournées sont toujours les bonnes après la modification. Dans le cas contraire, soit si la précision se retrouve réduite, la modification est annulée.

INDUS est totalement adapté aux conditions particulières des flux de documents (la forte hétérogénéité du flux, l'inégalité de représentation des classes, les faibles connaissances à priori et l'apparition subite de nouvelles classes). Cependant, il souffre des défauts hérités de A2ING (Manque de performance, pas de gestion de son expansion dans le temps et exige un opérateur humain) et il est fortement pénalisé en cas d'erreurs OCR sur des mots-clés.

En plus de ces méthodes, il existe des tentatives de créer des réseaux neuronaux incrémentaux, nous verrons de quoi il en retourne dans la prochaine section.

2.4.2 Apprentissage profond incrémental

Les structures classiques de réseaux neuronaux sont inadaptées à l'apprentissage incrémental, notamment dû à "l'oubli catastrophique" provoqué par le réentraînement. Cependant, il existe dans l'état de l'art plusieurs tentatives d'adaptations conçues pour permettre d'utiliser les réseaux dans des cas de flux séquentiels. Il est possible d'en dégager deux axes d'approches : soit se concentrer sur l'adaptation de la structure du réseau, soit utiliser des méthodes d'entraînement spéciales permettant de compenser les défauts du réentraînement.

Pour commencer, l'adaptation structurelle consiste à permettre au réseau de s'agrandir de manière incrémentale par une augmentation du nombre de paramètres à chaque nouveauté. Une des approches propose de diviser l'incrémentation en tâches rassemblant chacune plusieurs classes [63]. L'objectif devient dès lors de permettre d'entraîner le réseau sur une nouvelle tâche sans perdre en performance sur les tâches précédemment apprises. La solution repose sur des contrôleurs permettant de réadapter le réseau convolutif profond d'origine depuis la tâche 1 vers la tâche 2 en suivant la Figure 2.23. Pour cela, le contrôleur utilise une matrice de poids qui une fois appliquée aux poids de la couche les réadapte pour la nouvelle tâche (une matrice par tâche, sauf pour la première qui a les poids originaux). Cette matrice est entraînée alors que les autres poids du réseau sont figés. Il y a un contrôleur par couches de convolution permettant ainsi de toutes les modifier. La fin du réseau (couches denses) est par contre séparée en plusieurs classifieurs, un par tâche. Un système de décision (pouvant être un sous-réseau) permet de faire le choix des paramètres et du classifieur à utiliser en fonction de la donnée d'entrée.

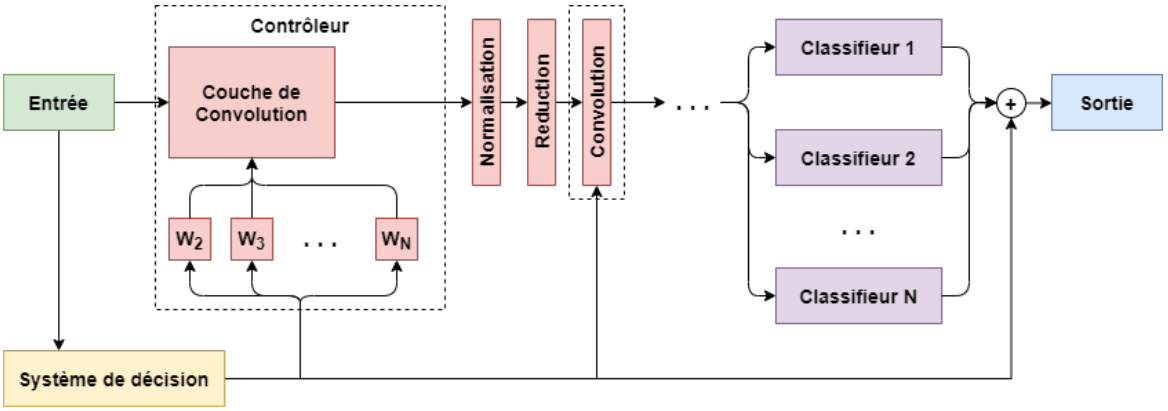

FIGURE 2.23 – Architecture proposée pour l'adaptation structurelle d'un réseau convolutif profond à l'apprentissage incrémental [63]

Pour intégrer les nouvelles informations qui arrivent au fur et à mesure au réseau neuronal, il est nécessaire de le réentraîner. La solution privilégiée pour réduire le nombre de réentraînements nécessaires est de rendre l'apport de nouvelles informations séquentielles. En conséquence, le réseau reste fixe le temps que suffisamment de nouveaux exemples et/ou de nouvelles classes soient disponibles. Cependant, dû au problème d'oubli catastrophique, le réentraînement ne peut se faire simplement à partir des poids du réseau obtenus à la fin de l'entraînement précédent sous peine de perdre les connaissances déjà acquises. Pour contrer cet effet, il est possible de mettre en place une méthode d'entraîne-

ment spécifiquement adaptée.

L'une des options possibles est de ne réentraîner que la partie finale du réseau [67]. Cette méthode nécessite de dupliquer seulement les dernières couches de convolution et les couches denses de classification (pour ajouter les nouvelles classes). Le reste du réseau (les couches de convolutions précédentes) est partagé entre les deux branches. La partie conservée maintient ainsi les connaissances déjà acquises et la partie clonée doit permettre de s'adapter aux nouvelles classes et exemples. Cependant, un équilibre est à trouver entre les deux. En effet, plus la partie conservée est grande, moins le réseau s'adapte à la nouveauté.

Une autre option est de chercher à compenser directement le problème du réentraînement [40]. Cette méthode cherche là aussi à entraîner le réseau avec une incrémentalité séquentielle, de manière à ce que l'entraînement de la séquence t ne provoque pas de perte de connaissance par rapport à celui de la séquence $t-1$. Pour ce faire, cette méthode utilise l'inférence Bayésienne pour estimer la distribution à posteriori des séquences en fonction des paramètres du réseau (les poids). Cette distribution est estimée par inférence stochastique variationelle [30].

Les méthodes d'adaptation actuellement présentes dans l'état de l'art sont encore trop récentes et expérimentales pour une utilisation directe dans notre situation. Celles-ci exigent toujours un réentraînement à chaque nouveauté et bien qu'elles cherchent à réduire le coût, il reste plus élevé que pour la plupart des autres méthodes d'apprentissage incrémental. Ce point contraint à un apprentissage plus séquentiel qu'incrémental, les documents ne pouvant être pris en compte qu'en groupe et non un par un, forçant à assembler un grand nombre de documents de la nouvelle classe avant de pouvoir la prendre en compte. Ces méthodes restent finalement très proches du transfert de connaissance (voir Section 2.1.3). Ce champ de recherche reste néanmoins à surveiller car des évolutions pourraient rendre ces méthodes intéressantes dans notre contexte.

2.5 Conclusion sur l'état de l'art

Le tableau 2.1 compare les différentes méthodes présentées précédemment sur les points importants du contexte (les réseaux neuronaux classiques ont déjà été traités avec la Figure 1.1).

TABLE 2.1 – Comparaison entre les méthodes d'apprentissage avec peu d'exemples et incrémentaux (**avantages**/**inconvénients** par rapport aux besoins)

Critères	**Renforcement & Augmentation**	**Zero-shot**	**One/Few-shot**	**Neural gas**	**Inc. Network**
Contraintes industrielles					
Vitesse d'exécution	✚	?	✚	━	✚
Coût Puissance de calcul	━	? (━)	━	✚	━
Compréhensibilité des erreurs	━	━	━	✚	━
Connaissance à priori nécessaire ?	✚	━	✚	━ (Pour le choix des caractéristiques utilisées)	✚
Adaptation au flux					
Classification des grandes classes	? (✚)	?	?	━	?
Classification des petites classes	?	?	✚	✚	?
Résistance au déséquilibre	✚	? (━)	✚	✚	━
Résistance à l'incomplétude	━	✚	━ / ✚ (selon les méthodes)	✚ (Capable d'ajouter de nouvelle classe à la volée, mais nécessite une intervention humaine ━)	━
Intégration de nouveauté	━	━	━ / ✚ (selon les méthodes)	✚	✚

En somme, l'état de l'art propose une grande variété de solutions spécifiques, mais aucune n'est idéale. Les réseaux neuronaux de classification classiques sont performants sur la majorité du flux (soit les grandes classes), s'exécutent rapidement et nécessitent pas beaucoup de connaissances à priori. Cependant, ils requièrent beaucoup d'exemples de résultats et ne s'adaptent pas vraiment au flux de documents. Les solutions d'apprentissage renforcé et d'augmentation des données semblent proposer des solutions permettant de compenser le problème du déséquilibre, mais il reste les problèmes des très petites classes

et de l'incomplétude. Le zéro-shot nécessite des connaissances à priori qu'il est considéré comme impossible à avoir dans notre contexte, ce qui le classe malheureusement comme hors sujet. De l'autre côté le one/few-shot learning semble une meilleure solution, bien qu'elle ne résolve pas forcément la problématique liée à l'incomplétude. Enfin, les solutions incrémentales proposent une solution à cela, mais sacrifient en contrepartie les performances sur les plus grosses classes. Les réseaux incrémentaux sont mis de côté, car ils s'adaptent moins bien au flux que les autres méthodes incrémentales, avec des résultats très incertains.

Le tableau 2.1, soulève néanmoins un problème majeur avec la présence de nombreuses zones grises (symbolisées par un point d'interrogation) sur les lignes "Classification des grandes classes" et "Classification des petites classes" : l'état de l'art manque d'un comparatif entre ces méthodes dans le cadre des flux de documents. Le chapitre suivant sera donc dédié à établir un comparatif entre les méthodes les plus pertinentes de l'état de l'art avec un protocole permettant d'évaluer leurs performances sur des flux de documents. Ces méthodes sélectionnées sont les suivantes :

— Pour les réseaux neuronaux de classification textuelles : un biRNN [69] avec BERT [19], une combinaison des deux méthodes de réseaux convolutifs [38, 85] (sous l'acronyme de TCNN) et la méthode de RCNN [44].
— Pour les réseaux neuronaux de classification d'image : VGG16 [70], HCNN [27] et ResNet [28] (qui malheureusement a été écarté pendant les premières phases de tests car il avait de mauvais résultats sur les bases privées de Yooz).
— Pour les méthodes d'apprentissage sur peu d'exemples : Prototypical Network [71] (ProtoNet) et NTM [66]. Ces deux solutions ont été sélectionnées car elles offraient des possibilités d'améliorations pour résoudre le problème de l'incomplétude. Cependant, à l'instar de ResNet, NTM à été mis de côté pour les mêmes raisons.
— Pour les méthodes d'apprentissage : A2ING [11] (choisi car ayant déjà montré son efficacité sur des flux de documents et y étant déjà adapté).

Chapitre 3

Protocole d'évaluation et comparatif de l'état de l'art dans le contexte des flux de document

Avant de commencer le développement de nouvelles méthodes, il a fallu trouver un moyen d'évaluer celles de l'état de l'art qui montraient un potentiel de compatibilité avec notre problématique. L'évaluation de ces méthodes porte sur leurs performances globales, avec un accent sur la précision, et sur leur aptitude à les conserver sur les petites classes du flux. Si elles ne peuvent pas les classer correctement, alors il faudra au moins qu'elles les rejettent. La capacité de la méthode à pouvoir intégrer de nouvelles classes est également un plus non négligeable. Pour cela, il faudrait pouvoir évaluer l'adaptation des méthodes retenues aux contraintes des flux de documents avec des corpus suffisamment larges pour les entraîner et les tester. Les corpus utilisés doivent également rester dans le domaine des documents industriels et administratifs proches de ceux traités par Yooz.

Cette partie contient les informations associées à l'évaluation des méthodes de l'état de l'art dans le contexte des flux de document. Elle commence par la présentation des corpus et bases de données utilisés. Elle se poursuit avec la description du protocole de test, ce qui inclut la sélection des mesures et les tests d'adaptations aux contraintes des flux. Elle se termine par les résultats des expériences et leur conclusion.

3.1 Corpus

Pour tester et évaluer les différentes méthodes retenues à partir de l'état de l'art et les solutions développées deux bases de données distinctes ont été sélectionnées. Ces deux bases correspondent à une base industrielle privée caractéristique des conditions de la thèse et à une base scientifique publique permettant une comparaison avec l'état de l'art. La base publique a été choisie sur des critères de tailles et de proximités avec notre objet de recherches. Ces critères exigent de disposer d'assez d'exemples pour que les résultats soient les plus statistiquement fiables et rester le plus proche de notre sujet de documents d'entreprises que possible. La base privée doit elle permettre d'analyser les méthodes dans des conditions proches de leur contexte d'utilisation finale. Les performances des méthodes sur cette base sont donc primordiales.

La première base, appelée YoozDB, compte 23 532 documents en français répartis inégalement sur 47 classes. Les classes comptent entre 1 et 3397 documents dont la distribution suit la figure 3.1. Les classes comptent parmi elles de multiples variations de factures, de bons de commande, de relevés bancaires, d'avis d'imposition, de mails, de chèques, de cartes d'identité, ... Cette liste n'est pas exhaustive mais donne une bonne idée des types de documents présents (Pour la liste complète voir figure 3.2). Cette base rassemble d'authentiques documents provenant des clients de Yooz avec les images scannées et le texte extrait via ABBYY-FineReader (version 14.0.107.232). L'ensemble d'entraînement compte 15 491 documents (65.71%), l'ensemble de validation 2203 (9.34%) et celui de test 5883 (24,95%). Dans l'ensemble, le corpus YoozDB est constitué comme un flux de documents d'entreprises classique qui illustre parfaitement le domaine d'application finale de la thèse.

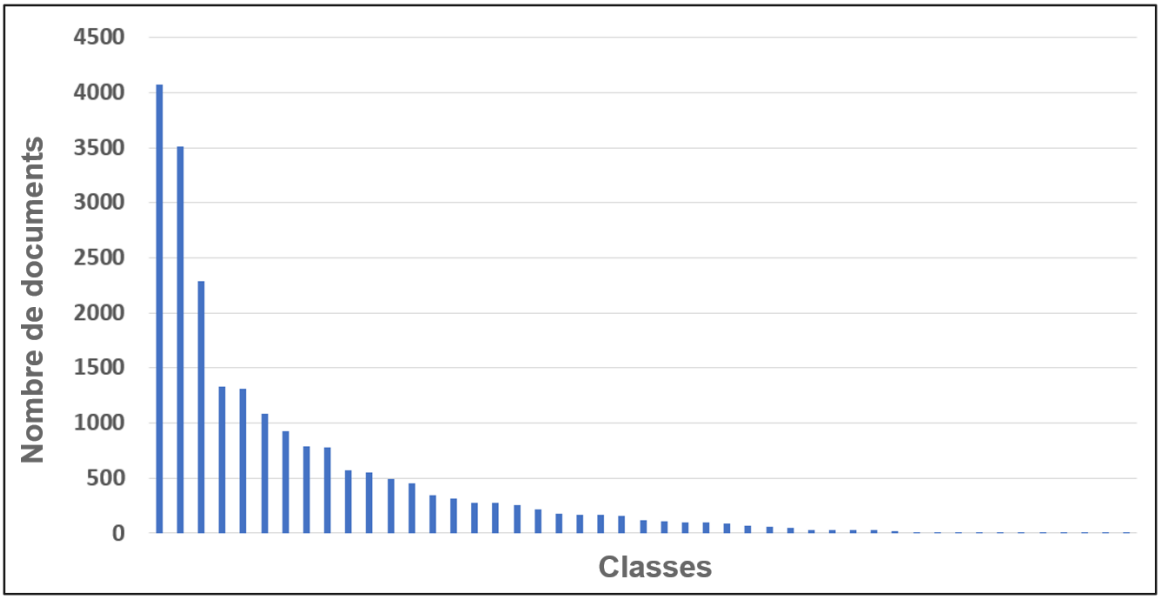

FIGURE 3.1 – Distribution des documents par classe du corpus YoozDB

ACNT_INVOICE	BANK_STATMNT	BUSN_GDRCPT	BANK_CREDITSTATMNT	BUSN_STATMNT	BANK_BES	ACNT_DUES
ACNT_EXP	ACNT_PAYSLIP	ACNT_INVOICELIST	BUSN_DELIVNOTE	BUSN_ORDER	BUSN_QUOTE	ACNT_CALLCAPITAL
BANK_ACCOUNTID	ACNT_TAXNOTICE	ACNT_NOTICEDEBIT	BANK_PAYABLENOTE	LGAL_CONTRACT	ACNT_NOTICEPAY	ACNT_STATMNT
CGV	BUSN_CONTRACT	ACNT_DUCS	ACNT_NOTIFICATION	ACNT_FINALDEMAND	INSUR_CONTRACT	ACNT_FINE
MAIL_BANK_PAYABLENOTE	BUSN_WORK	CHQ	MAIL	LGAL_REPORT_GA	LGAL_WORKSTOP	VERSO
LGAL_KBIS	ACNT_DEBITMANDATE	INSUR_DAILYALLOWANCE	BANK_DEBITSTATMNT	ID	MAIL_ACNT_DUES	MAIL_RETURNCOUPON
BANK_OPPOSITION	LGAL_CV	MAIL_ACNT_INVOICE	MAIL_BUSN_CONTRACT	MAIL_ACNT_CALLCAPITAL		

FIGURE 3.2 – Liste des 47 classes du corpus YoozDB (même ordre que pour la distribution)

Cette base pose cependant un problème majeur lié à sa forme de flux de document : son ensemble de test. En effet, celui-ci est aussi incomplet et déséquilibré que le reste du

corpus. Cette propriété permet certes d'évaluer les performances globales des différentes méthodes dans des conditions similaires à celles où elles seront utilisées. Cependant, en contrepartie, elle ne permet pas du tout de tester réellement l'efficacité des méthodes sur les petites classes, que nous voulons justement ne pas ignorer. Même si les systèmes utilisés sont incapables de traiter les plus petites classes, la distribution de l'ensemble de test leur permet d'afficher d'excellentes performances en ne classant exclusivement que les types de documents les plus représentés. Une représentation des performances classe par classe, ainsi qu'une évaluation par validation croisée montrent que les méthodes sont en moyenne moins bonnes sur les plus petites classes que sur toutes les autres (Pour plus de détails, voir section 3.3.1). Cependant, la quantité de données est trop faible pour qu'il soit possible d'obtenir des résultats statistiquement fiables sur l'efficacité réelle des méthodes évaluées pour les classes les moins représentées. En effet, celle-ci ne sont représentées dans l'ensemble de test que par un ou deux exemples, voire aucun. De plus, les documents de ces classes sont souvent très similaires plus parce qu'ils proviennent du même fournisseur que parce qu'ils sont représentatifs de la diversité interne réelle de ces classes. En conséquence, les résultats ne permettent pas de comparaison fiable entre les méthodes sur ce point et ce même avec une validation croisée.

Afin de pallier ces problèmes, nous proposons d'utiliser une seconde base, RVL-CDIP, qui comprend 400 000 documents provenant de l'industrie du tabac. Les documents sont majoritairement en anglais et répartis équitablement sur 16 classes (voir Figure 3.3). Celle-ci est un sous-ensemble du corpus IIT-CDIP qui standardise ces images sur un format 754*1000 pixels, 72 dpi, mono-page, là où les originaux sont de tailles variables, d'une résolution de 200 ou 300 dpi, en binaire et potentiellement multipages. Les documents sont répartis comme suit : 320 000 documents pour l'apprentissage, 40 000 pour la validation et 40 000 pour le test.

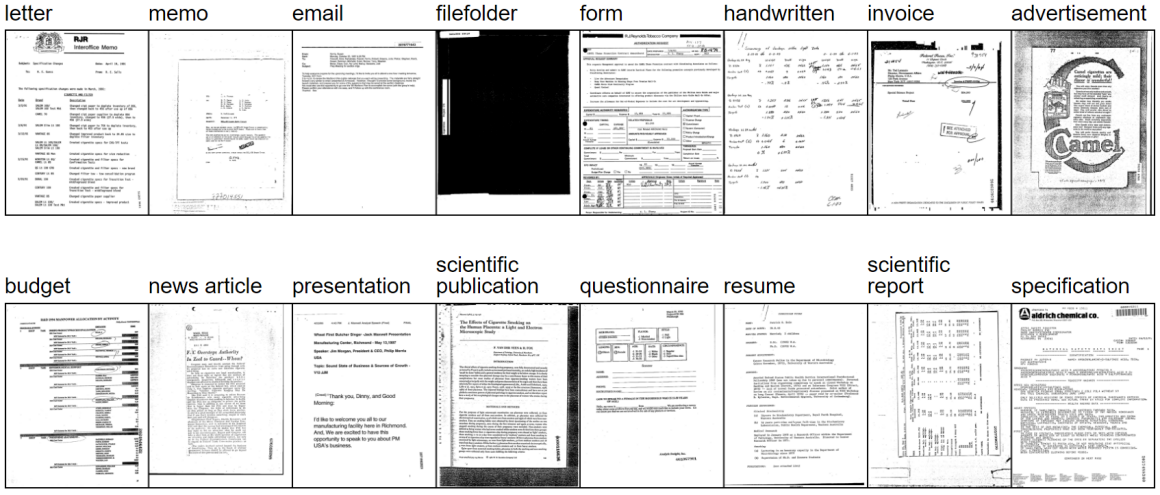

FIGURE 3.3 – Présentation des classes de RVL-CDIP

Afin d'évaluer convenablement toutes les méthodes, il a fallu compléter le corpus RVL-CDIP avec la partie textuelle de la base. Celle-ci est cependant problématique car organisée de manière différente, car regroupée par documents selon l'organisation de IIT-CDIP et non de RVL. Cela fausse un peu les résultats puisqu'il y a plus d'informations dans la base textuelle qui ne sont pas réellement présentes sur les images. Pour résoudre ce

problème et profiter des avancées en matière d'OCR des dernières années, le corpus a été complété avec la version originale de chaque image provenant de IIT-CDIP (offrant une meilleure qualité de résolution). Un fichier contenant le texte extrait par un nouvel OCR sur ces images a été ajouté pour chaque image. L'OCR est plus récent que l'original datant de 2006 et est équivalent à celui utilisé sur les images de la base YOOZDB. Cette opération permet de disposer d'une base plus complète, contenant l'image et le texte organisé de manière équivalente. Dans les deux cas avec la meilleure qualité possible, afin de comparer les deux approches sur un pied d'égalité. Le nouvel OCR permet également de prendre en compte l'évolution technique de ce domaine et les images d'origine offrent la possibilité de mieux choisir les prétraitements à appliquer.

Dans l'ensemble, le corpus est plus conçu comme une base de données d'images (au détriment de la partie textuelle), comme le montrent les classes "handwritten" et "file folders" qui sont des classes totalement inadaptées à la classification de texte. Ces deux classes sont presque totalement dépourvues de mots, ce qui force à les reconnaître sur cette absence et donc rend difficile de les distinguer entre elles. Ces classes n'ont pas d'équivalent d'un point de vue de l'image, ne permettant pas de rééquilibrer les deux modalités. De plus, certains documents ne sont pas en anglais mais trop peu nombreux pour que cela appuie une forme de multilinguisme du corpus, ce qui passe donc plus pour du bruit qu'autre chose.

En somme, ces deux bases de données servent des objectifs différents et sont complémentaires l'une de l'autre. La base YoozDB permet de tester les différentes méthodes sur un corpus proche du domaine d'application de la thèse. Ce corpus offre donc un moyen de sélection des méthodes en fonction de leur efficacité bien plus fidèle à notre thématique que RVL-CDIP. Cet autre corpus, quant à lui, convient bien plus à des évaluations de performances statistiquement fiables dans des cas de simulation de déséquilibre et d'incomplétude grâce à son ensemble de test et sa grande quantité de documents. De plus, RVL-CDIP permet une comparaison avec l'état de l'art.

3.2 Protocole de test

3.2.1 Mesures

Pour évaluer les performances d'un système de classification, il existe un grand nombre de mesures possibles, dont une donne partie est liée à la notion de matrice de confusion. Il s'agit d'une matrice sur laquelle sont reportées les prédictions en fonction de s'ils correspondent ou non à la vérité-terrain (l'étiquette de l'échantillon). Cette matrice (Figure 3.4) permet de calculer quatre valeurs :

— les "Vrais Positifs" (VP), qui rassemble tous les cas où la prédiction s'accorde avec la vérité-terrain sur un cas positif.
— les "Faux Positifs" (FP), qui correspond à l'ensemble des cas avec une prédiction positive mais une vérité-terrain négative.
— les "Faux Négatifs" (FN), qui est associés aux cas de prédictions négatives avec une vérité-terrain positive.
— les "Vrais Négatifs" (VN), qui compte chaque situation avec une prédiction et une vérité-terrain négative.

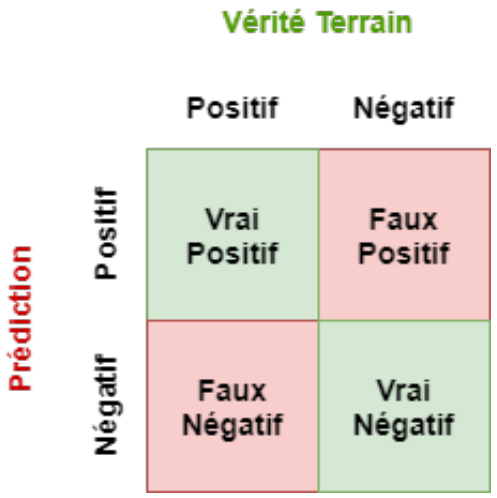

FIGURE 3.4 – Matrice de confusion binaire

Dans notre cas, il y a plus de deux classes possibles ce qui introduit une complexité supplémentaire. En effet, dans une situation multi-classes les notions de positif et de négatif sont relatives à la classe considérée et perdent donc tout leur sens d'un point de vue global, comme l'illustre la première matrice de la Figure 3.5. De fait, si on ne considère pas les classes séparément le nombre de FP et de FN sont égaux, les FP d'une classe étant les FN d'une autre. En conséquence, beaucoup des mesures calculées à partir de la matrice deviennent équivalentes, elles perdent donc beaucoup de leur intérêt. Par contre, ces mesures conservent leurs significations lorsqu'une seule classe est analysée à la fois, en suivant la deuxième matrice de la Figure 3.5. L'ajout d'un système de rejet perturbe encore cette modélisation des résultats si on souhaite intégrer les rejets dans la matrice de confusion, comme l'illustre la Figure 3.6. Dans ce cas, les rejets sont des FN en toute situation car la nouvelle classe des "rejetés" n'a pas de vérité terrain.

Pour rappel, les FN sont considérés dans notre contexte comme des erreurs moins

graves que les FP car ils sont catégorisés par le système de classification comme de potentielles erreurs. Cette mise en avant permet de leur dédier un processus de correction, avec une intervention humaine par exemple. Là où les FP se retrouvent mélangés aux autres et donc risquent de passer inaperçus. Pour cette raison, toutes les méthodes évaluées se voient adjoindre un système de rejets à seuil fixe basé sur le taux de confiance renvoyé lors de la classification de chaque document. Cette méthode de rejet est simple, mais elle a l'avantage de pouvoir s'appliquer à toutes les méthodes facilement, sans nécessiter de modifications qui pourraient altérer les résultats. Le taux de rejet est fixé expérimentalement pour chaque méthode, en étant le plus élevé possible comme l'exige le contexte (voir section 1.1).

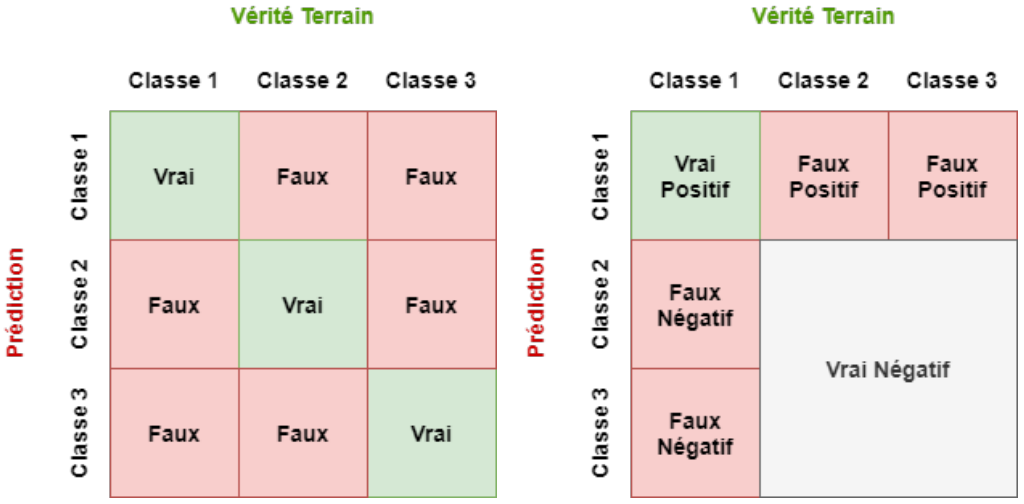

FIGURE 3.5 – Matrice de confusion multi-classes (ici trois). La première image montre lorsqu'on considère l'ensemble des classes (niveau global) et la second quand on ne considère que la classe 1 (niveau classe)

La plus commune des mesures issues de la matrice de confusion est sans doute "l'accuracy", qui pourrait être traduite par exactitude ou justesse. Elle est très utilisée dans l'état de l'art et correspond au pourcentage de bonnes réponses sur l'ensemble des prédictions. Son intérêt est de chiffrer les performances générales d'un système. Son calcul suit l'équation 3.1.

$$Accuracy = \frac{VN + VP}{FN + FP + VN + VP} \tag{3.1}$$

La précision évalue la proportion de bonnes réponses parmi l'ensemble des réponses positives. Elle permet notamment de s'assurer que lorsque le système évalué prédit, sa réponse est juste. En conséquence, plus celle-ci est élevée, moins il y a d'erreurs de classification. Si la précision est de 1, alors pour chaque prédiction d'une classe, celle-ci était la bonne. Le calcul de la précision se fait via l'équation 3.2. Son défaut, est qu'elle ne permet pas de savoir la proportion d'éléments classés, c'est pourquoi elle est souvent complétée par le rappel.

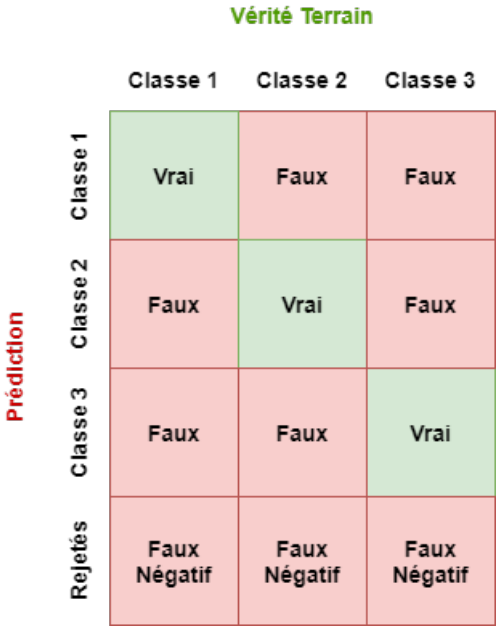

FIGURE 3.6 – Matrice de confusion multi-classes avec rejet

$$Precision = \frac{VP}{FP + VP} \tag{3.2}$$

Le rappel représente le degré de recouvrement, soit le pourcentage d'échantillons correctement reconnus parmi l'ensemble possible. Si celle-ci est de 1, cela signifie que tous les échantillons d'une classe ont été correctement placés dans celle-ci. Ce qui ne permet pas de savoir si des échantillons d'autres classes n'ont pas été classés par erreur dans celle-ci, d'où sa complémentarité avec la précision. Le calcul du rappel suit l'équation 3.3.

$$Rappel = \frac{VP}{FN + VP} \tag{3.3}$$

La F-mesure est une combinaison qui permet de rassembler en une seule mesure la précision et le rappel. Elle peut se calculer comme une moyenne harmonique des deux, on parle dans ce cas de F1-Mesure. Le paramètre β permet de pondérer l'importance de la précision par rapport au rappel, concrètement si celui-ci est supérieur à 1 alors le poids de la précision diminue et inversement si il est inférieur à 1. La F-mesure se calcule par l'équation 3.4.

$$F\beta - Mesure = (1 + \beta^2)\frac{Precision \times Rappel}{(\beta^2 \times Precision) + Rappel} \tag{3.4}$$

Il faut ajouter à cela d'autres mesures potentiellement pertinentes dans notre cas : le taux de rejet et taux d'erreur. Soit le rapport nombre de rejets sur nombre total d'échan-

tillons dans l'ensemble et nombre d'erreurs sur nombre total. Dans la situation de la première matrice de la Figure 3.5 : Accuracy = Précision = Rappel = F-Mesure.

Dans l'ensemble beaucoup de ces mesures sont redondantes entre elles selon le niveau de détail que l'on regarde. Pour conserver un maximum d'informations pour un minimum de mesures au niveau des résultats globaux, nous avons retenu trois mesures majeures : la précision, l'accuracy et le taux de rejet. La précision est la plus importante puisque c'est elle que l'on cherche à maximiser, comme précisé dans le contexte (section 1.1). L'accuracy permet une meilleure comparaison avec l'état de l'art, car elle est la mesure la plus courante. Le taux de rejet constitue un bon complément aux deux pour s'assurer que l'on ne sacrifie pas trop de Vrais Positifs pour maintenir la précision élevée. Le rappel est plus adéquat pour l'analyse classe par classe (en complément de la précision), cependant au niveau global elle est redondante avec l'accuracy (qui lui sera alors préféré). À noter que pour toutes ces mesures, les rejets sont considérés comme des Faux Négatifs dans les calculs.

La F0,5-Mesure est aussi un bon moyen d'évaluation permettant la synthèse des différentes valeurs (nous avons utilisé pour fixer le seuil de rejet). Elle peut être également un moyen de comparaison absolu entre deux méthodes qui serait difficile à départager avec trois mesures différentes. La pondération de la F-Mesure en faveur de la précision est motivée par le contexte. Quant au choix de la valeur de β, elle vise à approcher un ratio imposé de 4 :1 (4% d'accuracy/rappel équivaut à 1% de précision).

3.2.2 Tests d'adaptation

Pour évaluer l'efficacité des solutions potentielles face aux contraintes des flux de documents, détaillées dans la section 1.2, nous avons mis en place trois types de tests d'adaptation. Ces tests sont respectivement : l'adaptation au déséquilibre, l'adaptation à l'incomplétude et la combinaison des deux, que nous appellerons ici "l'adaptation réaliste". Pour les réaliser, il faut cependant que l'ensemble de test utilisé soit complet et équilibré. Cela permet de s'assurer autant de la fiabilité des résultats d'un point de vue statistique, que de celle de la représentation de la diversité interne des classes. Les modifications apportées par les tests d'adaptation à RVL-CDIP ne s'appliquent donc que sur les ensembles d'entraînement et d'évaluation. Cela permet de simuler les conditions d'un flux, tout en gardant l'ensemble de test tel quel assurant que les tests soient toujours faits sur les mêmes documents (permettant la comparaison).

Le test d'adaptation au déséquilibre répartit équitablement les seize classes en quatre groupes. Ces groupes correspondent à ceux que l'on peut établir à partir de la distribution du flux, présentés dans l'introduction (Section 1.2 et illustré par la Figure 1.1). Le premier est le groupe de classes sur-représentées et qui composent la majorité du flux. Le second est le groupe des classes moyennement représentées dont la quantité de document par classe est suffisante pour l'entraînement d'un réseau mais tout de même deux fois inférieure à celle du premier groupe. Le troisième est le groupe de classes faiblement représentées qui disposent d'à peine assez de documents pour le définir et qui donc commence à poser des problèmes pour les réseaux neuronaux. Enfin, le dernier est le groupe des classes qui ne sont représentées que par quelques documents, si ce n'est un seul. Ce groupe correspond aux conditions des défis du “few-shot learning" défini dans la sec-

tion 2.3.2. Chacun de ces quatre groupes est donc associé à une quantité de documents qui les représente par rapport aux autres groupes. Pour conserver cet aspect dans le test d'adaptation, les groupes sont chacun associés à un pourcentage. Ces pourcentages fixés, de 100%, 50%, 10% et 5%, sont des modificateurs appliqués sur la quantité de documents à conserver dans les ensembles d'entraînement et de validation pour une classe donnée. Par exemple, dans le cas de RVL-CDIP, une classe du groupe 1 conserve 20 000 pour le représenter en entraînement et 2500 en validation, tandis qu'une classe du groupe 2 conserve 10 000 en entraînement et 1250 en validation. La répartition des classes entre les groupes, ainsi que la sélection des documents conservés pour représenter les classes, est faite de manière aléatoire pour éviter tous potentiels biais liés à l'expérimentateur.

L'objectif de ce test est de vérifier l'adaptation du système évalué uniquement au déséquilibre et de permettre une analyse des effets que celui-ci a sur les résultats, ainsi que la forme qu'ils prennent. Les méthodes efficaces dans un contexte déséquilibré ne devraient pas être trop impactées à l'inverse de celles qui ne s'y prêtent pas du tout. Le choix de répartir les classes équitablement entre les groupes est là pour limiter au maximum l'impact des spécificités d'une classe sur l'ensemble du groupe, même si cela ne correspond pas à la réalité d'un flux de document (l'idéal aurait été d'avoir plus de classes). Sur ce test, il est attendu des méthodes qu'elles conservent leurs précisions sur l'ensemble des classes, quitte à ce que le taux de rejet augmente même fortement. Le degré de réussite du test se fait donc sur la conservation ou non de la précision et sur son coût au niveau de l'accuracy (conséquence de l'augmentation du taux de rejet).

Le test d'adaptation à l'incomplétude consiste à séparer équitablement en deux groupes les seize classes. Le premier de ces groupes rassemble les classes complètes qui correspondent aux classes majeures sur lesquelles il faut que la méthode évaluée conserve ses performances. Le second est le groupe des classes "incomplètes" qui sont alors représentées par un seul exemple (le premier rencontré). Ici aussi les classes sont réparties équitablement pour minimiser leurs impacts car la répartition entre les groupes est aléatoire, au même titre que l'exemple choisi pour représenter chaque classe incomplète.

Il s'agit du test le moins informatif des trois pour ce qui est des performances des méthodes évaluées, mais son objectif est autre. Il est en effet plus conçu pour analyser les effets spécifiques de ce type de classe sur les méthodes évaluées. Il est attendu de la méthode évaluée qu'elle conserve autant que possible ses performances sur les classes complètes et de rejeter les échantillons liés aux classes incomplètes. Si la méthode arrive à classer correctement quelques éléments des classes incomplètes, alors il s'agit d'un plus non négligeable pour ce test. Dans l'ensemble, ce test cherche plus à évaluer la capacité de la méthode testée à bien délimiter le périmètre des connaissances acquises et à reconnaître puis s'adapter à la nouveauté.

Le test d'adaptation réaliste combine les deux précédents, il s'agit du véritable test ayant pour but d'évaluer pleinement l'adaptation aux flux de documents. Ce test reprend la répartition des classes en groupes du déséquilibre mais sur 12 classes (soit 3 classes par groupe), puis ajoute le groupe des classes incomplètes (soit 4 classes incomplètes). Les affectations sont toujours aléatoires et dans le même esprit que précédemment.

Pour réussir ce test, la méthode évaluée doit conserver de hautes performances sur les plus grandes et moyennes classes. Elle doit également maintenir une précision élevée sur les petites classes avec le taux de rejet le plus bas possible. Enfin, dans le cas des

classes incomplètes la méthode doit s'assurer qu'elles n'impactent pas les performances des autres classes en rejetant ses échantillons pendant la phase de test. Un plus notable si la méthode parvient à reconnaître quelques documents de ces classes.

3.3 Résultats

3.3.1 Corpus brut et contexte équilibré

Les résultats de l'évaluation des méthodes sur les corpus originaux sont résumés dans le Tableau 3.1. Il y a déjà, un élément saillant dans la colonne correspondant à la base privée de Yooz, le réseau biRNN montre des performances excellentes bien que celle-ci soit un extrait de flux de documents. La précision est très élevée (supérieure à 99%) et l'accuracy (comme le taux de rejet) indique que presque tout le corpus a été classé. Les autres méthodes s'en sortent moins bien et globalement les méthodes usant de caractéristiques textuelles sont meilleures sur ce corpus, comme attendu au vu de sa composition. Les méthodes de classification d'images (VGG et HCNN) affichent néanmoins des performances surprenantes, dépassant ce qui pouvait être attendu au regard de la composition de YoozDB qui rassemble beaucoup de documents à l'apparence similaire mais aux classes différentes. VGG est ici très légèrement supérieur à HCNN. Le réseau ProtoNet et le système A2ING conservent des résultats honorables mais clairement inférieurs à leurs concurrents. Les méthodes NTM et ResNet ont également été testées sur ce corpus, mais ont été rapidement écartées car les réseaux ne parvenaient pas à converger. Ces résultats montrent une efficacité globale plutôt bonne des réseaux neuronaux sur les flux de documents, cependant ceux-ci sont en bonne partie dus aux particularités du corpus (comme décrit dans la section 3.1). Pour mieux comprendre, il faut entrer dans le détail des classes en les analysant selon leur représentation dans le flux.

TABLE 3.1 – Résultats sur les corpus originaux (sans modifications)

Corpus	YoozDB			RVL-CDIP		
Methodes / Mesures	Accuracy	Précision	Taux de Rejet	Accuracy	Précision	Taux de Rejet
biRNN	**93.57%**	**99.22%**	**5.69%**	77.95%	88.58%	**12.01%**
TCNN	88.36%	98.97%	10.72%	65.52%	93.72%	30.09%
RCNN	77.66%	98.13%	20.86%	50.80%	86.10%	41.00%
A2ING	79.07%	97.88%	19.22%	22.58%	**95.94%**	76.46%
VGG16	84.31%	96.12%	12.29%	80.24%	92.70%	13.44%
HCNN	84.57%	95.67%	11.61%	**80.76%**	95.42%	15.36%
ProtoNet	63.56%	97.93%	35.10%	63.68%	91.36%	30.30%

Le Tableau 3.2 contient les résultats par groupes de classes organisés selon le nombre de documents disponibles pour l'entraînement de chaque classe, à noter qu'il n'y a que les trois méthodes qui ont affiché les meilleures performances de leurs catégories (biRNN pour le texte, VGG16 pour l'image et ProtoNet pour le few-shot learning). Ce tableau illustre bien la problématique posée par base privée de Yooz au niveau de son ensemble de test : les plus petites classes, qui sont les plus difficiles, sont trop peu représentées dans l'ensemble de test pour influencer les performances globales. Il révèle également que le groupe le plus difficile (<50) est celui où l'ordre d'efficacité s'inverse. Le biRNN est le plus mauvais des trois et le Prototypical Network affiche la meilleure précision. Dans l'ensemble les méthodes sont surtout très efficaces sur les classes les plus représentées (groupe 1 et 2), mais celles-ci perdent en précision et en rappel lorsque le nombre d'échantillons d'entraînement par classes descend en dessous de 500 (autour de -10% dans les deux mesures pour VGG16 et biRNN). Le ProtoNet souffre moins mais il ne

parvient pas à dépasser clairement VGG16 et biRNN. Sur le plus petit groupe (<50), ProtoNet notablement plus précis (+6%), mesure la plus importante dans notre contexte.

TABLE 3.2 – Résultats par groupe de classes sur YoozDB

Nombre de Classes	3	6	14	6	18
Échantillons	>=1000	]1000;500]	]500;100]	]100;50]	<50
biRNN - YoozDB					
Précision	98.23%	97.27%	91.10%	89.20%	64.22%
Rappel	98.64%	97.84%	90.07%	88.02%	57.99%
VGG16 - YoozDB					
Précision	92.84%	89.23%	80.54%	83.45%	66.94%
Rappel	95.01%	89.76%	76.98%	81.11%	61.81%
ProtoNet - YoozDB					
Précision	90.20%	84.42%	79.45%	82.69%	72.92%
Rappel	89.76%	87.73%	72.86%	65.24%	52.38%

Pour la partie RVL-CDIP du Tableau 3.1, les méthodes de classification issues de l'image se démarquent plus que pour YoozDB. HCNN se montrent légèrement plus performants que VGG, là aussi contrairement à YoozDB, et qui est alors la méthode montrant les meilleurs résultats. VGG est un peu en dessous des performances présentées dans [3], ce qui doit venir des poids d'initialisation ou de la bibliothèque utilisée pour l'implémentation qui sont les seules différences. Toutes les méthodes utilisant le texte semblent souffrir de la faible quantité de mots par documents (particulièrement pour deux classes) et affichent des résultats mitigés. Là aussi biRNN semble un peu meilleurs que les trois autres, bien que sa précision laisse à désirer par rapport à TCNN. La méthode de A2ING est ici complètement dépassée, le peu de mots et les problèmes d'OCR lui empêchant par bien des aspects de s'entraîner convenablement. À noter qu'elle parvient tout de même à atteindre une précision importante malgré ces difficultés. Pour finir le réseau ProtoNet parvient à égaler les méthodes de réseaux profonds textuels, bien qu'il ne soit pas conçu pour la classification de grandes quantités.

La disparité de performances entre les réseaux textuels et les réseaux images sur RVL-CDIP s'explique par la difficulté de certaines classes qui ne sont tout simplement pas prévues pour une classification par le texte comme les "File Folder" (séparateur de dossier) et les "Handwritten" (document manuscrit). Ces deux classes ne contiennent presque aucun mot reconnaissable par un OCR (si ce n'est aucun tout court) et donc ne peuvent presque pas être différenciées par cette modalité. Ceux-ci sont clairement visibles sur le Tableau 3.3 pour les résultats avec biRNN (les autres méthodes textuelles offrent des retours suivant également la tendance décrite). D'autres classes comme "Advertisement" (publicité) et "Présentation" (diapositive) sont également assez difficiles pour ces méthodes avec peu de mots exploitables, ce qui se ressent là aussi sur les résultats. Par contre, les classes "Email" et "Resume" (curriculum vitae) se montrent beaucoup plus adaptées à cette stratégie, riche en mots et usant de termes spécifiques. Les méthodes visuelles retournent des forces et des faiblesses sur différentes classes, les résultats visibles sur le tableau sont ceux de VGG16 (mais là aussi HCNN et ProtoNet sont équivalents sur les points cités). Les classes "File Folder" et "Handwritten" sont ici bien plus simples à trai-

ter contrairement aux classes "Form" (formulaire), "Scientific Report" et "Questionnaire" qui elles se prêtent mieux aux méthodes textuelles. En somme, deux modalités semblent pouvoir se compléter si elles sont combinées. Cependant, la classe "Presentation" reste difficile à classer pour les deux modalités car elle contient peu de texte et a une diversité interne très élevée. Soit beaucoup de formes différentes pouvant se confondre avec les autres classes proches.

TABLE 3.3 – Comparaison par classe entre les méthodes textuelles et visuelles sur RVL-CDIP (Les valeurs de rappel n'incluent pas les rejets dans leurs calculs).

Méthode	biRNN							
Classes	**Letter**	**Form**	**Email**	**Handwr**	**Advert**	**Sc Report**	**Sc Public.**	**Speci**
Précision	89.38%	85.75%	96.81%	71.18%	85.86%	84.09%	90.12%	95.46%
Rappel	85.23%	83.71%	95.78%	76.46%	82.87%	**83.60%**	91.10%	**94.99%**
Taux de Rejet	**8.72%**	**9.40%**	**2.48%**	41.20%	18.52%	**10.24%**	**8.28%**	**3.48%**
Classes	**File F**	**News A**	**Budget**	**Invoice**	**Presen**	**Questi**	**Resume**	**Memo**
Précision	69.51%	88.00%	**90.77%**	92.00%	83.03%	**92.01%**	**98.89%**	91.86%
Rappel	80.08%	88.15%	89.95%	**92.78%**	82.31%	**90.66%**	**98.29%**	91.35%
Taux de Rejet	40.56%	**8.16%**	**6.88%**	**6.32%**	**11.80%**	**7.48%**	**1.96%**	**6.64%**

Méthode	VGG							
Classes	**Letter**	**Form**	**Email**	**Handwr**	**Advert**	**Sc Report**	**Sc Public.**	**Speci**
Précision	**92.19%**	**86.74%**	**98.41%**	**94.69%**	**92.33%**	**86.03%**	**94.92%**	**96.27%**
Rappel	**91.58%**	**84.24%**	**99.14%**	**97.40%**	**93.59%**	81.81%	**92.79%**	94.15%
Taux de Rejet	13.72%	20.99%	3.22%	**7.39%**	**13.12%**	28.26%	9.95%	8.05%
Classes	**File F**	**News A**	**Budget**	**Invoice**	**Presen**	**Questi**	**Resume**	**Memo**
Précision	**92.74%**	**92.61%**	90.43%	**92.81%**	**84.74%**	91.13%	97.58%	**94.86%**
Rappel	**98.70%**	**91.54%**	**92.38%**	92.18%	**88.19%**	89.13%	96.38%	**94.32%**
Taux de Rejet	**8.79%**	12.14%	16.73%	11.18%	24.11%	21.44%	7.53%	8.91%

Pour conclure, les réseaux profonds semblent montrer des performances satisfaisantes sur YoozDB (similaire à celles attendues dans un contexte équilibré). Cependant, cela n'est en réalité valable que pour les classes les plus représentées, les petites posent plus de problèmes car la quantité de documents d'entraînement est trop faible. Les méthodes textuelles sont nettement plus efficaces que celles issues du traitement de l'image. Pour RVL-CDIP, c'est l'inverse. Les méthodes images dépassent clairement les méthodes textuelles, principalement à cause de deux classes mais plus globalement dues aux faibles nombres de mots exploitables par documents. Les deux modalités semblent se compléter sur plusieurs classes et ainsi pouvoir offrir des résultats intéressants en cas de combinaison.

3.3.2 Tests d'adaptation

Les résultats sur les tests d'adaptation au déséquilibre confirment la tendance observée sur l'analyse par groupe de classes de YoozDB. Les classes les plus représentées parviennent à obtenir des scores élevés contrairement aux plus petites. Cependant grâce à l'ensemble de test équilibré et complet d'autres phénomènes deviennent visibles. Ceux-ci touchent à la gestion des classes déséquilibrées par les réseaux. Dans cette situation,

le système de rejet devient impératif pour conserver une précision relativement élevée car beaucoup d'erreurs proviennent des classes du groupe 5% (et 10% dans une moindre mesure). Les échantillons de ces classes sont captés par celles du groupe 100% qui sont plus représentées et donc mieux entraînées par le réseau. Ce phénomène ressemble à du sur-apprentissage, mais n'était pas présent lors du test avec le corpus d'origine, il est donc bien apparu avec le déséquilibre. La captation des échantillons n'est pas toujours uniforme entre les classes du groupe 100%. En effet, lorsque des classes avec une forte variété interne se trouvent dans ce groupe, celles-ci captent alors la majorité des échantillons. Elles deviennent ainsi des sortes de "classes poubelles" et plus cette variété est élevée, plus l'effet ce renforce. En cas d'équivalence de la variété interne des grandes classes, la répartition des erreurs s'équilibre entre les grandes classes.

Le Tableau 3.4 rassemble les résultats de toutes les méthodes pour la version déséquilibrée de RVL-CDIP (générée en suivant le processus décrit dans la Section 3.2). La mesure TdRM suivie du nom d'un groupe correspond au Taux de Rejet Moyen sur ce groupe. L'ensemble des méthodes perdent autour de 10% de leur accuracy, ainsi que quelques points en précision. Les exceptions notables sont ProtoNet qui ne semble tout simplement pas affecté par cette modification et VGG16 qui souffre particulièrement du déséquilibre. Le TdRM semble augmenter quand la représentation décroît. Les exceptions s'expliquent toutes par des particularités liées aux classes difficiles (voir Section 3.3.1) réparties au sein des groupes, sauf pour le ProtoNet qui n'est pas affecté par le déséquilibre et qui ne prend donc en compte que la difficulté des classes. Les "vainqueurs" de ce test sont donc HCNN et ProtoNet.

TABLE 3.4 – Résultats avec RVL-CDIP Déséquilibré (TdRM X : Taux de Rejet Moyen du groupe X)

Corpus	Déséquilibré						
Méthodes/Mesures	Accuracy	Précision	T. de Rejet	TdRM 100%	TdRM 50%	TdRM 10%	TdRM 5%
biRNN	**68.37%**	79.73%	**14.25%**	23.03%	4.03%	15.56%	14.39%
TCNN	51.98%	**91.63%**	43.27%	15.58%	58.53%	42.99%	55.98%
RCNN	40.71%	78.76%	48.31%	28.35%	35.43%	46.64%	84.85%
A2ING	—	—	—	—	—	—	—
VGG16	59.17%	88.90%	33.44%	18.68%	15.15%	53.36%	46.58%
HCNN	67.97%	89.14%	23.75%	8.60%	14.37%	30.31%	41.74%
ProtoNet	62.55%	90.60%	30.96%	27.80%	30.09%	40.46%	25.50%

Comme attendu pour le test d'adaptation à l'incomplétude (Tableau 3.5), les résultats sont assez mauvais. Comme la moitié des classes sont "incomplètes", le score maximal à espérer de ce test est 50% (pourcentage d'échantillons des classes entraînées). Si l'accuracy des réseaux est plutôt bonne (car proche de 50%), la précision des réseaux est catastrophique puisqu'elle peine à dépasser les 60%. Ceci démontre que les réseaux n'arrivent pas à distinguer correctement les classes qu'ils pouvaient entraîner (classes complètes) des autres. Le TdRM montre que les réseaux rejettent bien plus d'échantillons des classes "incomplètes", mais c'est très insuffisant avec 55,84% comme meilleurs résultats. La répartition des erreurs entre les classes "complètes" suit un principe proche de celui observé pour l'adaptation au déséquilibre. Les classes les plus faciles (comme "Resume" pour les réseaux texte) conservent de très hautes performances là où les plus difficiles ("file folder" pour les réseaux texte) se transforment en classes "poubelles". Elles se mélangent avec les

classes non entraînées. Là aussi la facilité semble correspondre à la diversité de la classe, les classes simples pour le texte étant des types de documents dont le vocabulaire est très spécifique et pour l'image des documents avec un format particulier presque identique d'un échantillon à l'autre. Le seul à avoir reconnu quelques documents d'une classe incomplète est ProtoNet, les autres ont une accuracy de 0% à ce niveau. Il n'y a aucun réseau qui montre des résultats satisfaisants, ce qui est un peu décevant pour ProtoNet mais reste prévisible car cette méthode reste plus adaptée au "five-shot" qu'au "one-shot".

TABLE 3.5 – Résultats avec RVL-CDIP Incomplet

Corpus	Incomplet				
Méthodes / Mesures	Accuracy	Précision	Taux de Rejet	TdRM Complète	TdRM Incomplète
biRNN	42.08%	49.82%	15.53%	9.55%	21.52%
TCNN	36.89%	61.38%	39.91%	23.97%	55.84%
RCNN	32.66%	47.40%	31.11%	26.19%	36.03%
A2ING	—	—	—	—	—
VGG16	42.12%	55.83%	24.56%	9.22%	39.91%
HCNN	43.54%	60.75%	28.34%	9.46%	47.22%
ProtoNet	40.22%	57.82%	30.44%	13.21%	47.68%

Les résultats du test d'adaptation réaliste (Tableau 3.6) sont moins bons que ceux du test de déséquilibre, ce qui n'est pas étonnant puisqu'il est plus difficile avec deux classes fortement déséquilibrées en plus et l'introduction des classes "incomplètes". Les observations faites sur le déséquilibre se retrouvent ici aussi avec l'apparition de classes "poubelles". Cependant, les performances sont meilleures que celles du test incomplet, notamment par l'augmentation du TdRM sur ce type de classe. Dans l'ensemble les résultats sont à mi-chemin entre ceux du test incomplet et du test de déséquilibre, ce qui reste finalement cohérent avec son objectif. Les méthodes perdent toutes autour de 25% de leur accuracy et 20% de leur précision par rapport à la version originelle du corpus. La seule exception est ProtoNet qui finit même par dépasser en performances HCNN et VGG16, bien que les classes "incomplètes" semblent être le principal problème pour cette méthode.

TABLE 3.6 – Résultats avec RVL-CDIP Réaliste

Corpus	Réaliste							
Méthodes / Mesures	Accuracy	Précision	Taux de Rejet	TdRM 100%	TdRM 50%	TdRM 10%	TdRM 5%	TdRM Incomplète
biRNN	50.71%	67.56%	**24.94%**	4.66%	11.53%	29.41%	29.13%	42.53%
TCNN	36.88%	78.50%	53.02%	22.83%	33.08%	63.87%	66.79%	72.16%
RCNN	28.74%	68.45%	58.02%	18.16%	48.12%	59.16%	89.29%	71.02%
A2ING	11.85%	**84.63%**	86%	78.19%	77.81%	85.20%	87.85%	97.20%
VGG16	51.28%	77.97%	34.24%	8.69%	17.48%	39.44%	32.80%	60.97%
HCNN	55.70%	77.55%	28.18%	6.81%	10.35%	29.13%	30.01%	55.49%
ProtoNet	**56.76%**	78.16%	27.38%	11.71%	17.67%	19.17%	16.32%	57.93%

Pour conclure, les résultats des tests d'adaptation montrent que les réseaux sont bien moins adaptés aux difficultés du flux que les résultats globaux de YoozDB ne le laissaient supposer. Il confirme le pressentiment qui ressort de l'analyse classe par classe et l'ag-

grave même. Certaines classes semblent être plus faciles à classer selon la modalité qui est utilisée, faisant varier les classes utilisées comme "poubelle" (même si certaines restent difficiles dans les deux cas comme la classe "Presentation"). Ce constat laisse penser que la multimodalité peut possiblement aider en rendant plus simple l'apprentissage de ce type de classes, diminuant *de facto* la confusion qu'elles génèrent. Cette option sera analysée plus en profondeur dans le chapitre 4. Les réseaux les plus courts (biRNN, TCNN et ProtoNet) semblent un peu mieux s'adapter que les réseaux plus profonds.

ProtoNet semble d'ailleurs être le réseau le moins impacté par les contraintes, mais comme ses résultats initiaux sont moins bons, il peine à dépasser les autres réseaux basés sur l'image. RCNN est le réseau dont les résultats sont les plus mauvais, la qualité de l'OCR et le faible nombre de mots exploitables n'y sont sûrement pas étrangers. C'est un réseau conçu en se reposant sur la globalité du texte plus que sur des mots-clés, contrairement à TCNN et biRNN qui n'ont finalement besoin que de peu de mots pour fonctionner (ce qui est un avantage certain). D'ailleurs, biRNN est globalement un peu meilleur que TCNN sur YoozDB et dans les pires cas au moins équivalant à celui-ci. A2ING n'est pas adapté à RVL-CDIP, ce qui rend l'analyse de son adaptation aux tests très difficiles, même s'il semble aussi affecté que les autres par les contraintes du test réaliste. Pour les réseaux visuels, le choix entre HCNN et VGG16 est assez difficile, d'un côté HCNN est meilleur sur RVL-CDIP autant sur le brut que sur les tests d'adaptation, de l'autre VGG16 est un peu meilleur sur YoozDB et est plus rapide à entraîner. La structure de VGG16 est également plus simple (un seul réseau contre 5 pour HCNN) ce qui le rend plus facile à modifier pour y tester des méthodes d'adaptation. Finalement, cela fait un peu plus d'avantages pour VGG16.

Pour la suite des expérimentations, les méthodes retenues sont donc biRNN, ProtoNet et VGG16. A2ING est mis de côté pour les évaluations sur RVL-CDIP, mais garde son intérêt sur YoozDB (Principalement pour l'aspect incrémental et la haute précision de l'algorithme).

3.3.3 Tests sur des données limitées

Pour compléter les tests d'adaptation précédents sur les méthodes les plus intéressantes. Une série d'évaluations avec des données de plus en plus limitées a été lancée pour analyser le comportement des réseaux dans cette situation. L'objectif est de voir à partir de quelle quantité de documents, les réseaux retenus commencent à avoir des performances correctes et si la quantité nécessaire varie d'une classe à l'autre. Si la quantité nécessaire varie, l'idée est de savoir quel critère est responsable de cette variation. Pour ce test, les quantités de documents par classe ont été fixées de manière équilibrée et les documents conservés choisis aléatoirement.

Les résultats sont résumés dans la Figure 3.7 avec les modèles retenus à partir des tests précédents soit VGG16, biRNN-Bert et Prototypical Network (pour rappel cette méthode de few-shot se base sur l'image des documents). La courbe montre que l'ensemble des réseaux conservent leurs positions jusqu'à 2000 documents par classes avec des performances qui baissent légèrement pour VGG16 et biRNN. ProtoNet stagne globalement autour de son accuracy initial. Il gagne un peu sur les 10000 par classe, ce qui s'explique par le fait qu'il ne s'agit pas d'un réseau conçu pour être entraîné avec une grande quantité de documents. La quantité n'est pas forcément à son avantage voire peut-être néfaste. Au-delà de 2000 documents par classe, la donne change. Le réseau VGG16 va s'effondrer

rapidement tandis que biRNN est rattrapé par ProtoNet. Ces deux réseaux s'équivalent jusqu'à 50 documents par classe. À partir de là, le réseau biRNN va plonger plus rapidement que le réseau ProtoNet qui parvient à mieux s'adapter que les deux autres.

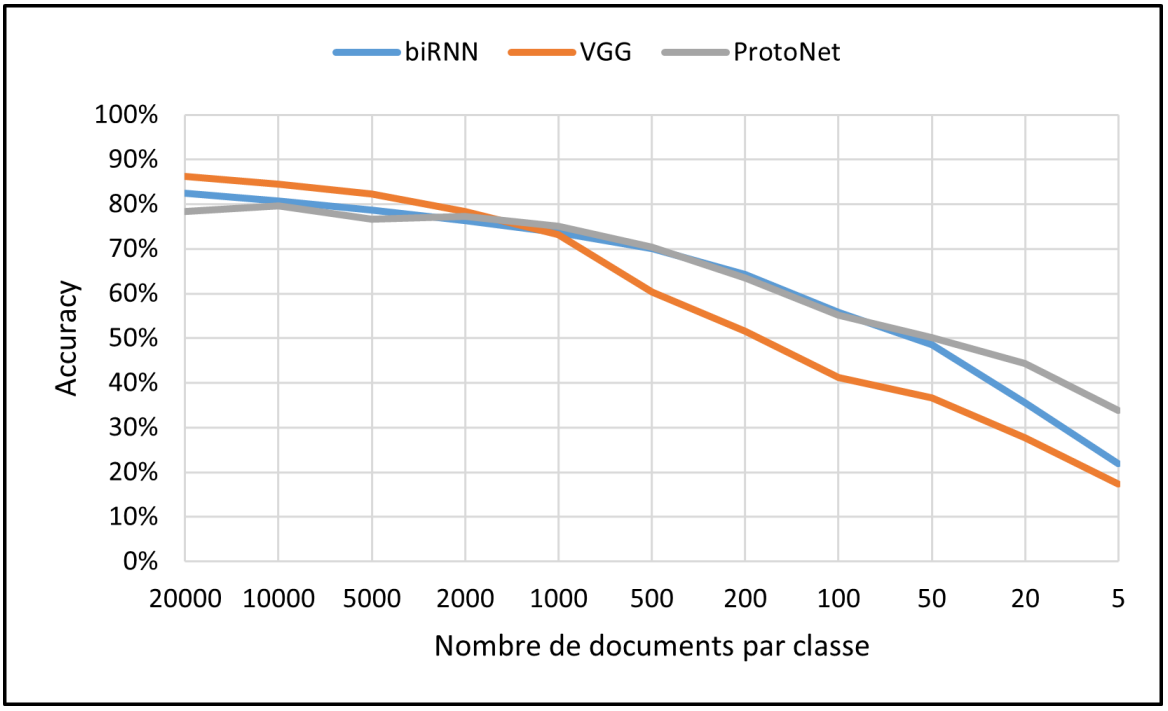

FIGURE 3.7 – Accuracy des méthodes en fonction du nombre de documents par classe

Dans l'ensemble, ces résultats confirment l'hypothèse qui se dégage de l'analyse classe par classe de YoozDB (Tableau 3.2), le réseau ProtoNet est plus efficace que biRNN et VGG16 sur les très petites classes. De plus biRNN a visiblement moins besoin de documents que VGG16 pour s'entraîner. Les résultats de VGG16 et biRNN étaient attendus, ce sont des réseaux conçus pour s'entraîner sur de vastes corpus et non sur de petites quantités d'exemples. Le réentraînement à partir d'ImageNet semble moins profiter à VGG16 que l'encapsulation Bert à biRNN, ce qui est peu surprenant. Un modèle de langue généraliste reste bien plus à propos qu'un pré-entraînement sur une base d'images naturelles (un peu hors sujet) pour de la classification de documents. La comparaison entre un pré-entraînement sur une base de documents et une base d'image sur ces résultats aurait été intéressante mais il n'existe pas pour le moment de base comparable à ImageNet dans le domaine de la classification de document (même RVL-CDIP reste assez petite en comparaison et est de toute façon inutilisable ici). Les résultats de ProtoNet sont finalement un peu décevants car ils ne dépassent clairement ceux de biRNN qu'en dessous de 50 documents par classes. Cependant, ProtoNet est deux fois plus efficace que les autres réseaux en "five-shots", bien que les performances restent faibles ce qui témoigne de la difficulté de classer correctement 40000 documents à partir de seulement 80 exemples.

Pour compléter cette analyse, le tableau de la Figure 3.8 affiche la précision et le rappel par classe pour trois des tests avec un ensemble limité. Il s'agit des tests pour 1000 documents par classe (5%), pour 100 documents par classe (0,5%) et pour 20 documents par classe (0,1%), avec en premier les résultats pour RVL-CDIP complet comme point

de comparaison. Ce tableau permet de remarquer qu'une fois de plus, la perte de performance due à la réduction du nombre d'échantillons, n'est pas la même en fonction des classes. La réduction étant uniforme pour toutes les classes, cette différence doit provenir de la classe elle-même. L'effet de la modalité utilisée par le réseau semble également présent à l'instar des tests d'adaptations. Pour l'image, les classes "email", "handwritten", "advertisement", "file folder" et "resume" sortent du lot. Tandis que pour le texte, c'est plutôt les "email", "scientific publication", "specification" et "resume". Il s'agit donc de classes "faciles" à différencier des autres en usant de cette modalité et pour la plupart de ces classes cette propriété se constate au niveau des échantillons. Deux classes sont redondantes car elles ont une grande similarité interne dans ce corpus autant au niveau du texte que de la forme. Les autres le sont seulement au niveau du texte ou de l'image et se complètent, l'usage d'un modèle multimodal pourrait tirer profit de ces classes "faciles" pour obtenir de meilleurs résultats même en cas de déséquilibre.

En somme, la modalité semble affecter la diversité interne et externe d'une classe au niveau des caractéristiques utilisables par le réseau. La diversité impacte la "facilité" de la classe à être apprise par le réseau malgré le manque de documents d'entraînement. Les classes "faciles" selon la modalité sont complémentaires sur plusieurs d'entre elles.

Label	letter	form	email	hand written	advert	scient report	scient publi	specif	file folder	news article	budget	invoice	present	question	resume	memo
RVL-CDIP - 100%																
biRNN																
Precision	85,71%	81,36%	95,77%	55,22%	78,98%	78,10%	87,55%	93,23%	63,94%	83,06%	87,37%	89,16%	78,10%	89,29%	97,78%	90,06%
Rappel	82,28%	78,20%	94,12%	76,44%	72,76%	77,44%	86,64%	92,52%	59,44%	83,92%	85,76%	89,48%	75,44%	85,68%	97,00%	87,72%
VGG16																
Precision	87,64%	76,30%	97,23%	90,42%	86,68%	74,30%	91,04%	92,17%	86,96%	88,00%	82,43%	87,62%	73,09%	81,57%	93,78%	90,38%
Rappel	84,62%	75,42%	97,54%	94,67%	87,99%	70,62%	86,90%	89,04%	95,77%	86,07%	84,83%	86,27%	78,99%	79,06%	91,53%	89,37%
ProtoNet																
Precision	76,98%	59,33%	97,52%	87,95%	84,18%	42,20%	90,99%	87,87%	78,53%	88,94%	73,17%	73,11%	67,50%	80,85%	96,57%	88,21%
Rappel	77,16%	63,48%	92,80%	92,56%	84,92%	55,92%	82,80%	79,72%	94,64%	76,84%	69,28%	74,40%	70,52%	70,24%	86,80%	81,72%

Label	letter	form	email	hand written	advert	scient report	scient publi	specif	file folder	news article	budget	invoice	present	question	resume	memo
RVL-CDIP - 5%																
biRNN																
Precision	74,43%	66,34%	86,98%	50,26%	67,32%	64,74%	82,05%	88,36%	56,98%	74,07%	75,80%	77,99%	64,24%	79,75%	93,54%	81,51%
Rappel	68,24%	67,32%	88,20%	69,84%	62,20%	63,60%	78,24%	87,76%	48,64%	77,48%	74,44%	80,24%	62,24%	78,00%	95,00%	76,52%
VGG16																
Precision	73,40%	55,56%	92,08%	82,30%	78,72%	51,23%	82,63%	82,93%	80,12%	77,97%	66,25%	70,91%	56,07%	65,37%	83,53%	71,16%
Rappel	72,08%	53,20%	93,88%	91,12%	80,96%	49,00%	79,92%	77,56%	90,60%	75,04%	65,08%	66,40%	64,28%	56,48%	83,56%	72,84%
ProtoNet																
Precision	72,02%	51,49%	96,77%	87,04%	80,62%	37,69%	90,73%	86,62%	76,81%	86,68%	73,21%	72,40%	67,06%	78,11%	95,61%	85,29%
Rappel	71,96%	62,92%	90,00%	91,04%	83,36%	61,56%	81,80%	75,64%	93,16%	75,20%	62,40%	65,88%	65,80%	64,52%	85,32%	69,80%

Label	letter	form	email	hand written	advert	scient report	scient publi	specif	file folder	news article	budget	invoice	present	question	resume	memo
RVL-CDIP - 0,5%																
biRNN																
Precision	33,82%	40,89%	81,98%	44,45%	46,54%	42,84%	63,67%	77,07%	50,34%	53,88%	59,31%	48,60%	41,51%	67,38%	89,45%	59,81%
Rappel	34,56%	41,12%	80,08%	58,16%	44,72%	43,08%	59,72%	71,92%	50,20%	58,32%	54,68%	58,16%	35,88%	57,00%	86,80%	58,04%
VGG16																
Precision	38,00%	23,37%	71,54%	46,53%	32,31%	22,35%	48,76%	49,94%	65,95%	35,63%	34,04%	32,28%	35,55%	24,25%	52,11%	28,64%
Rappel	47,36%	26,00%	80,44%	52,60%	34,20%	18,44%	53,28%	53,28%	70,12%	29,20%	32,84%	25,64%	28,04%	21,44%	59,40%	25,44%
ProtoNet																
Precision	51,17%	29,22%	78,05%	68,64%	71,33%	23,23%	77,63%	63,22%	56,90%	67,63%	45,93%	41,36%	56,41%	31,65%	56,25%	51,20%
Rappel	39,20%	24,60%	84,76%	92,52%	77,84%	20,12%	71,76%	47,72%	91,72%	68,04%	46,68%	30,92%	43,32%	42,76%	75,20%	24,76%

Label	letter	form	email	hand written	advert	scient report	scient publi	specif	file folder	news article	budget	invoice	present	question	resume	memo
RVL-CDIP - 0,1%																
biRNN																
Precision	18,85%	15,75%	41,19%	34,86%	22,13%	22,28%	52,45%	45,13%	34,46%	45,37%	41,14%	28,80%	25,92%	30,77%	73,37%	33,73%
Rappel	24,36%	15,00%	36,92%	12,76%	19,64%	15,08%	47,04%	58,20%	88,08%	38,60%	25,08%	39,36%	11,24%	29,28%	78,24%	30,60%
VGG16																
Precision	20,55%	11,12%	44,37%	29,65%	51,78%	13,64%	16,05%	33,37%	38,68%	22,74%	25,72%	13,53%	19,55%	9,57%	29,02%	14,13%
Rappel	11,04%	9,92%	76,00%	42,76%	62,36%	12,96%	12,88%	26,20%	54,60%	23,76%	13,92%	17,56%	6,64%	2,88%	58,68%	10,12%
ProtoNet																
Precision	41,45%	20,36%	65,21%	55,79%	70,44%	17,16%	68,90%	49,88%	48,72%	54,29%	37,30%	19,46%	44,55%	15,45%	52,34%	24,64%
Rappel	27,44%	20,60%	71,24%	81,48%	64,52%	12,20%	70,44%	41,40%	89,24%	53,16%	28,56%	26,04%	24,84%	10,80%	64,04%	22,16%

FIGURE 3.8 – Résultats par classe avec RVL-CDIP 100%, 5%, 0.5% et 0.1% pour biRNN, VGG16 et ProtoNet *(sans rejet)*

3.4 Conclusion de l'évaluation

Les réseaux profonds offrent des résultats globaux élevés sur la base YoozDB, qui sert de référence de flux de documents. Les méthodes textuelles montrent des résultats clairement supérieurs à celles utilisant l'image. Ces résultats sont contrebalancés par une analyse classe par classe qui montrent que ces performances proviennent principalement des classes les plus représentées. Dès que celles-ci le sont moins, les performances baissent surtout pour les classes les plus petites. La quantité de documents permettant de tester à ce niveau reste cependant trop faible pour en ressortir des conclusions fiables. Sur RVL-CDIP, les méthodes visuelles sont plus efficaces que les méthodes de classification de texte contrairement à YoozDB. Les performances sont inégalement réparties entre les classes et dépendent de la modalité utilisée par le réseau.

Il ressort des tests d'adaptation que le déséquilibre est effectivement un sérieux problème pour les réseaux neuronaux, beaucoup plus que les tests sur YoozDB ne le laissaient supposer. Les méthodes plus adaptées aux petites classes sont peu ou pas affectées par le déséquilibre. Ce test conforte l'importance de l'utilisation d'un système de rejet pour conserver une précision assez élevée et l'intérêt des méthodes de "few-shot" pour les petites classes. L'incomplétude fait ressortir principalement que les réseaux ont tendance à confondre les échantillons des classes qui n'ont pu être entraînées avec ceux des classes avec une forte diversité interne et externe. Ces classes sont difficiles à entraîner pour bien les distinguer des autres. Le test réaliste combinant les deux (déséquilibré et incomplet) montre que les effets se cumulent et réduit encore les performances par rapport au déséquilibre seul.

Les résultats des tests avec une réduction du nombre d'échantillons d'entraînement par classes appuient l'importance de la diversité intra-inter classes et de la modalité. Ils montrent également que le réseau de "few-shot learning" devient clairement plus efficace que les autres réseaux à partir de 50 documents par classe ou moins et les égale à partir de 1000. Les méthodes retenues à l'issue des tests sont les réseaux VGG16, biRNN pour de la classification de grandes classes et ProtoNet pour la classification des petites.

Il ressort de cette évaluation la nécessité d'appliquer des méthodes pour renforcer la résistance des modèles de réseaux neuronaux sélectionnés contre le déséquilibre. En ce sens, la combinaison de ces modèles avec les propositions de l'état de l'art comme le renforcement de l'entraînement ou la génération d'exemples (détaillées dans la section 2.2) est clairement une option à explorer. De plus, le choix de la modalité affecte les performances en fonction des classes. La multimodalité s'impose donc également comme solution intéressante pour améliorer les performances globales autant que pour faciliter l'apprentissage des classes et la distinction entre elles. Il reste également, le cas des systèmes d'attention qui pourraient, au même titre que la multimodalité, renforcer encore les modèles sélectionnés en améliorant la qualité des caractéristiques qu'ils entraînent. Il nous faut construire des modèles utilisant ces différentes options ainsi qu'évaluer leur efficacité dans un contexte déséquilibré. Cette évaluation doit permettre de déterminer l'effet de chaque option individuellement, ainsi que les potentielles synergies qu'elles pourraient avoir entre elles.

Ensuite, il restera la problématique de l'intégration des classes incomplètes et de la

classification des classes très peu représentées (classes en few-shot). Les méthodes permettant de gérer ces classes et les méthodes les plus efficaces sur les grandes classes sont très différentes et spécialisées. La meilleure solution pourrait être de réussir à combiner ces différentes méthodes en essayant de tirer profit de leurs avantages respectifs. Ces différents éléments sont le sujet des prochains chapitres.

Chapitre 4

Multimodalité, adaptations et systèmes d'attention dans les cas déséquilibrés

Dans le chapitre précédent, les multiples évaluations ont démontré que les réseaux neuronaux les plus performants de l'état de l'art dans un contexte équilibré (image comme texte) s'adaptent mal au contexte fortement déséquilibré des flux de documents. Elles révèlent également que les performances des réseaux varient entre les classes en fonction de la modalité utilisée pour les entraîner.

Ce chapitre présente plusieurs solutions développées avec pour objectif d'améliorer les réseaux pour la classification de documents et/ou la classification déséquilibrée. L'idée est de renforcer les meilleures méthodes retenues après le chapitre 3 (VGG16 et biRNN) avec les propositions de l'état de l'art les plus récentes qui ont un potentiel pour améliorer au moins un des deux points précédemment cités. Dans ces propositions de l'état de l'art, se retrouvent la classification multimodale, les systèmes d'attention et les deux méthodes de renforcement contre le déséquilibre.

La multimodalité a le potentiel d'améliorer les performances globales des réseaux, comme discuté dans la section précédente. Les systèmes d'attention se dégagent de l'état de l'art comme étant une option d'amélioration des réseaux neuronaux, permettant d'augmenter l'accuracy comme la précision (voir section 2.1.5). Ils constituent donc une option de choix dans notre contexte si ces gains sont toujours présents dans des cas déséquilibrés. Pour finir la génération et le renforcement de l'entraînement sont les contre-mesures de l'état de l'art au déséquilibre (voir section 2.2). Il nous faut encore évaluer la fiabilité de ces méthodes dans notre contexte et si elles peuvent se combiner entre elles. L'objectif final de cette partie est de proposer un modèle de réseau neuronal, le plus adapté possible à la classification de documents fortement déséquilibrée.

4.1 Modèle multimodal

Pour résoudre le problème de la grande variété des classes existantes dans notre contexte, la multimodalité semble être une bonne option. La pertinence des modalités dépend des classes, même si pour le type de grandes classes dans notre contexte, le texte reste le plus efficace (voir section 3.3). La combinaison des deux modalités a pour objectif de permettre au réseau de maintenir des performances optimales sur toutes les classes, qu'elles soient plus simples à traiter à partir de son texte ou de son image. La combinaison peut aussi avoir un intérêt en cas de déséquilibre. Comme le montre la Section 3.3, la modalité influe

sur la facilité avec laquelle le réseau entraîne une classe. Plus une classe est homogène pour le réseau, moins celui-ci est affecté par le manque d'échantillons d'entraînement (les éléments disponibles sont alors suffisamment représentatifs des possibilités de la classe).

Nous avons testé deux propositions d'architectures de réseaux multimodales inspirées des travaux de Souhail et al. [7]. Les deux propositions combinent les réseaux VGG16 et biRNN(Bert), avec le premier pour la partie image et le second pour le texte. Il s'agit, pour rappel, des deux modèles de réseaux neuronaux classiques qui se sont montrés être les plus performants, pour leurs catégories respectives, dans le chapitre précédent (Chapitre 3).

Pour la première proposition la combinaison s'opère sur les couches de sorties qui sont concaténées entre elles. Le résultat de cette concaténation sert alors d'entrée à un nouveau classifieur dense en deux couches séparées par une de dropout, à l'instar des classifieurs utilisés par les réseaux eux-mêmes. L'ensemble de l'architecture suit dans sa globalité la Figure 4.1. D'autres méthodes de combinaison ont été essayées, comme la multiplication ou la somme, mais les résultats restaient très légèrement en faveur de la concaténation (bien que les trois donnent des résultats très proches et sont donc probablement équivalents). Cette architecture permet d'obtenir une combinaison simple permettant de prendre en compte les deux modalités dans la classification des documents, en utilisant le réseau le plus performant parmi ceux testés pour chaque modalité.

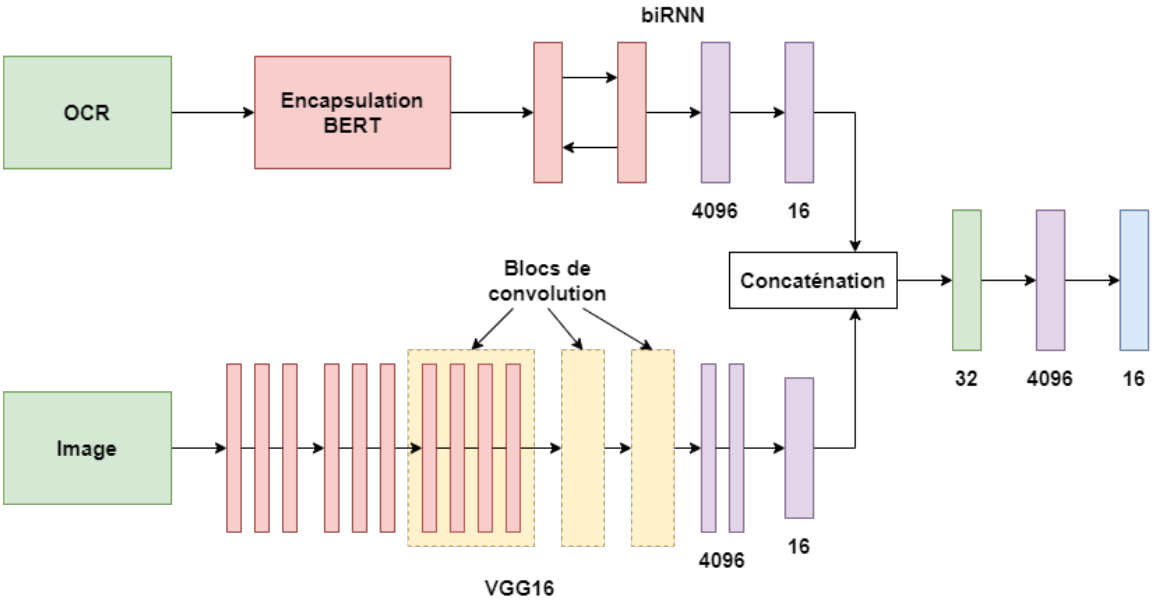

FIGURE 4.1 – Première architecture proposée

La seconde proposition d'architecture suit plutôt la Figure 4.2. La partie qui change le plus par rapport à la précédente est la combinaison qui utilise l'avant-dernière couche de chacun des deux réseaux plutôt que la dernière. De plus, à la place d'une concaténation, la combinaison se fait avec une addition (car là encore, il s'agit de celle qui offrait les meilleurs résultats bien que ce soit de très peu, à l'instar de Souhail et al. [7]). Cette version d'architecture multimodale cherche à combiner les caractéristiques extraites par les deux réseaux de base. Sur ce point, la première version cherche plus à utiliser le meilleur

des deux réseaux en fonction du document.

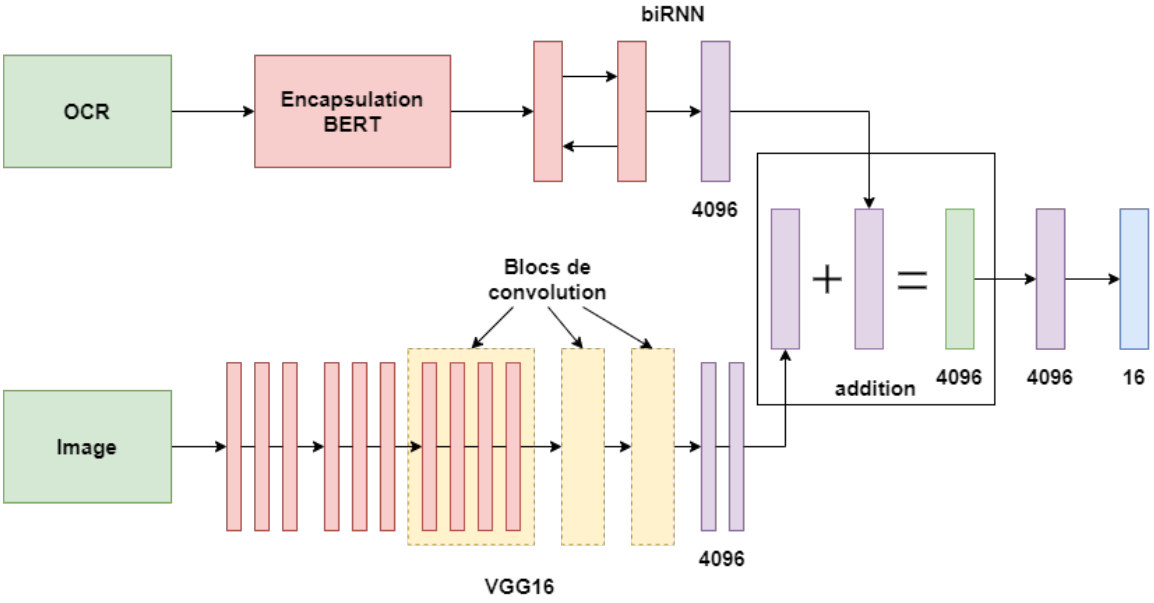

FIGURE 4.2 – Seconde architecture multimodale proposée

Les méthodes proposées utilisent le même processus d'entraînement. Dans un premier temps, chacun des deux sous-réseaux (biRNN/VGG16) est entraîné séparément. Une phase finale d'entraînement est alors opérée avec l'architecture multimodale permettant de régler les paramètres du méta-classifieur et d'affiner l'entraînement global du réseau. Durant cette phase finale les paramètres des modèles pré-entraînés séparément sont figés pour permettre de se concentrer sur les nouveaux. Cette méthode d'entraînement est employée car les autres options (entraînement du modèle multimodal sans poids pré-calculés ou sans fixer ces poids) ne permettent pas au réseau de converger quelle que soit l'architecture utilisée. Il semble que la fonction de coût ne parvient pas à faire face à la profondeur de l'architecture et au nombre de paramètres (sachant que le réseau VGG16 sans poids pré-entraîné sur ImageNet a déjà beaucoup de mal à s'entraîner, que ce soit sur YoozDB ou sur RVL-CDIP).

Après plusieurs tests sur ces deux architectures, la seconde version a été privilégiée car elle montrait clairement de meilleures performances sur RVL-CDIP (environ +1.16% d'accuracy et +3.22% de précision, en contrepartie de +1.87% de taux de rejet par rapport à la première version). Sur YoozDB les résultats des deux versions sont globalement équivalents. De plus, sur le plan théorique, la première architecture ne permet pas d'entraîner de véritables caractéristiques multimodales. Cela vient du fait que le méta-classifieur ne reçoit en entrée que les décisions produites par ses sous-réseaux et non les caractéristiques utilisées pour les prendre. Cette particularité n'a d'intérêt que si la combinaison des caractéristiques textuelles et visuelles est plus néfaste que bénéfique (dans le sens où elle porterait à confusion), ce qui serait en contradiction avec nos observations. En conséquence, l'architecture utilisée pour la suite de ce manuscrit en tant que modèle multimodal est la seconde version (Figure 4.2).

4.2 Systèmes d'attention

Les mécanismes d'attention ont un triple intérêt pour notre contexte. Le premier est qu'ils améliorent les performances des réseaux qui les utilisent, comme le démontrent les nombreuses publications récentes sur le sujet [19, 23, 33]. L'amélioration semble provenir d'une meilleure pertinence des caractéristiques utilisées par le réseau, car celui-ci est guidé par un système d'attention qui le focalise en écartant celles qui sont hors-sujet. Cette propriété de sélection est clairement recherchée pour la classification de documents où seules quelques zones où mots-clés précis sont déterminants pour en reconnaître le type.

Le second intérêt découle du premier, en effet les classes sous-représentées ont d'autant plus besoin d'une meilleure sélection des caractéristiques pour être convenablement entraînées. Cependant, cela nécessite que le système d'attention conserve cette propriété malgré la baisse du nombre d'échantillons disponibles en le compensant. Le dernier intérêt est celui de rendre plus compréhensibles les erreurs du réseau et donc de faciliter l'analyse de celles-ci, permettant notamment de voir concrètement quels mots ou zones sont utilisés comme caractéristiques pour une classe de documents donnée.

4.2.1 Modèle d'attention textuel

Le mécanisme utilisé pour le biRNN est un modèle d'attention simple inspiré des premiers [15]. Il se place à la fin de la partie récurrente du réseau, dont il utilise la sortie h comme entrée, une fois que celle-ci a été aplatie (noté h_f). Le module se compose d'une couche dense suivie d'une activation $softmax$ qui permet d'obtenir le vecteur d'attention t. Un *dropout* a également été ajouté pour éviter le sur-apprentissage (noté $drop()$). Il se termine par le produit scalaire entre le vecteur d'attention t et la sortie h des couches cachées de LSTM. Le résultat de ce produit A_t est alors envoyé à la partie de classifieur dense du réseau. L'ensemble est résumé par la figure 4.3 et l'équation 4.1.

$$A_t = \underbrace{softmax(drop(h_f))}_{\text{t}} \cdot h \tag{4.1}$$

4.2.2 Modèle d'attention visuel

Le mécanisme d'attention utilisé pour le réseau VGG16 est basé sur les travaux de Górriz et al. [23]. Il doit permettre au réseau de concentrer son attention sur les zones les plus importantes du document, comme celles contenant des titres, des logos ou des tableaux. Ce module d'attention utilise comme entrée un ensemble *h* de *N* feature maps provenant d'un des blocs de convolution de VGG16. Deux couches de convolution 1x1 suivent pour extraire des caractéristiques spatiales à partir de *h*. Ces caractéristiques sont ensuite envoyées à une dernière couche de convolution 1x1 avec des connexions localisées [13] et une fonction d'activation de type sigmoïde, générant ainsi le masque d'attention. Celui-ci est appliqué par une multiplication pixel à pixel sur les maps de *h*. Le résultat, noté *h'*, est ensuite réduit par un average pooling global (hérité de VGG16). Puis, il est relié à une couche dense disposant d'une fonction d'activation softmax chargée de la classification. Le module est résumé par la Figure 4.4.

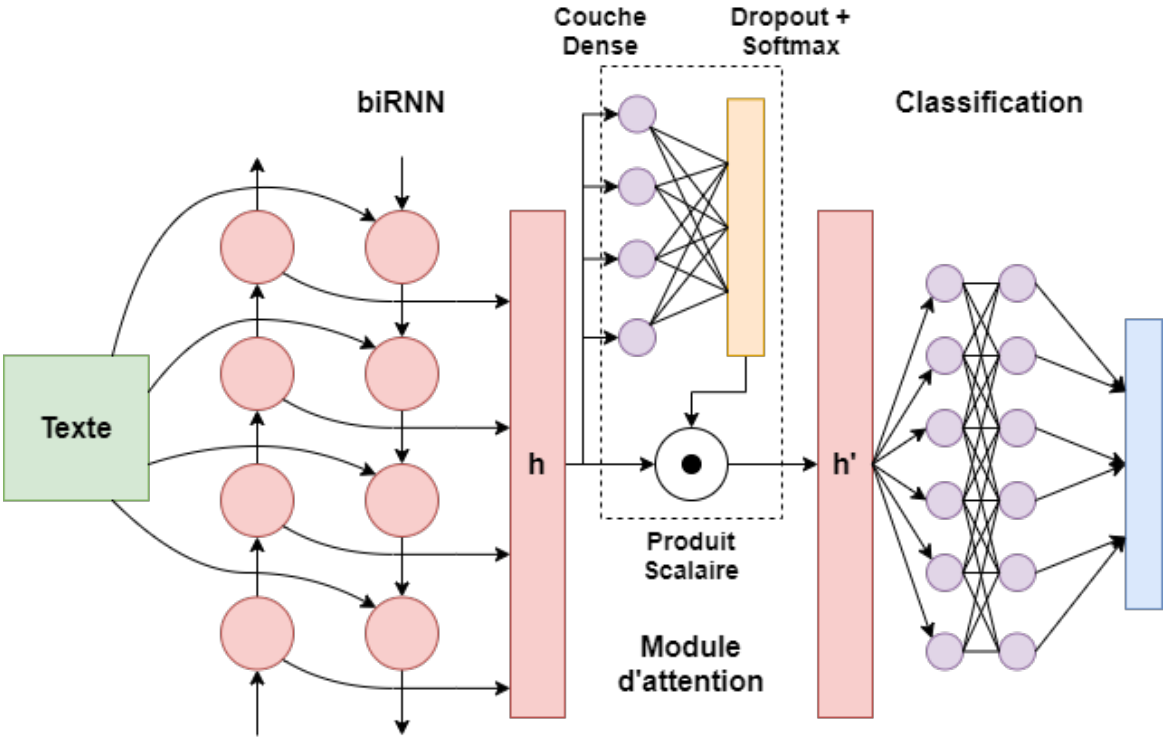

FIGURE 4.3 – Modèle d'attention textuel

Cependant dans notre situation, il y a des limites aux ressources disponibles pour entraîner les réseaux. Les réseaux profonds comme VGG16 sont déjà coûteux en termes de mémoire et en puissance de calcul. En conséquence, la méthode que nous utiliserons ne peut conserver qu'un mécanisme d'attention, sur les trois proposés par Górriz et al. [23] (originellement placé en sortie des blocs de convolution 3, 4 et 5). Ce problème est d'autant plus présent que l'un des objectifs est de tester un modèle multimodal (qui est lui aussi plus coûteux par l'ajout du deuxième modèle et de la combinaison). Le matériel à notre disposition ne permet simplement pas de tester des modèles plus gourmands en mémoire GPU. Le choix de la couche a été fait à partir d'expériences visant à déterminer le meilleur endroit où le placer.

Ces expériences sont résumées par le Tableau 4.1 avec les résultats sur la base privée de Yooz à gauche et à droite la base RVL-CDIP. Les lignes correspondent aux numéros des blocs de l'architecture VGG16 à partir desquels le réseau est tronqué (la note $VGG16_{bloc2}$ signifie que le réseau s'arrête au deuxième bloc sur les cinq ; l'ajout du $_{-att}$ indique la présence du modèle d'attention).

Comme les résultats sont très serrés sur l'ensemble des mesures, nous avons utilisé la F0,5-Mesure pour les départager. Pour rappel, cette F-Mesure renforce l'importance de la précision dans son calcul (voir section 3.2.1 pour plus de détails). Même avec la F0,5-Mesure, les résultats ne donnent pas de vainqueur évident, puisque le meilleur change entre YoozDB et RVL-CDIP et sont très proches. Pour choisir, nous avons donc pris la moyenne des F0,5-Mesures entre les deux corpus, afin de déterminer laquelle de ces options offre les meilleurs résultats globalement sur l'ensemble des deux corpus.

Pour la version sans attention, les moyennes des F0,5-Mesures sur les deux corpus

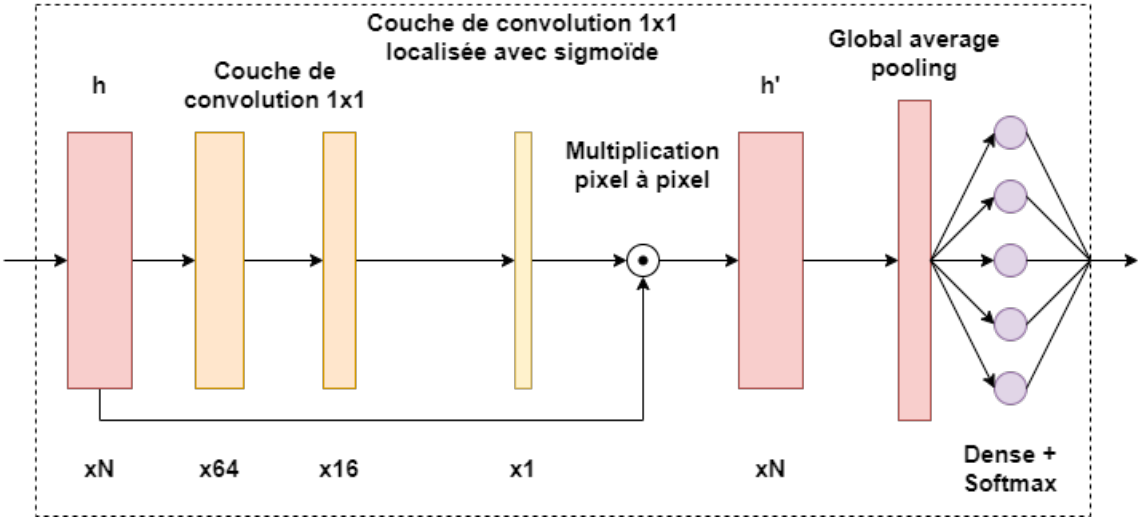

FIGURE 4.4 – Modèle d'attention visuel

sont respectivement : 90.53% pour $VGG16_{bloc2}$, 91.14% pour $VGG16_{bloc3}$, 90.74% pour $VGG16_{bloc4}$, **91.70%** pour $VGG16_{bloc5}$. Ces résultats donnent gagnante l'architecture allant jusqu'au bloc 5. Cependant, pour la version avec système d'attention, c'est celle allant jusqu'au bloc 4 qui offre les meilleurs résultats, avec des moyennes respectives de : 88.43% pour $VGG16_{bloc2}$, 93.30% pour $VGG16_{bloc3}$, **93.61%** pour $VGG16_{bloc4}$, 93.36% pour $VGG16_{bloc5}$. Puisque la version avec modèle d'attention qui nous intéresse le plus, l'architecture conservée est celle allant jusqu'au bloc 4. Celle-ci est légèrement plus efficace que celle allant jusqu'au bloc 5 tout en étant moins coûteuse (car plus petite).

TABLE 4.1 – Résultats sur les corpus originaux (sans modifications)

Corpus	YoozDB				RVL-CDIP			
Méthodes / Mesures	Accuracy	Précision	T. Rejet	F0,5 M.	Accuracy	Précision	T. Rejet	F0,5 M.
$VGG16_{bloc2}$	82.61%	**96.33%**	14.24%	93.23%	61.82%	**98.17%**	37.03%	87.84%
$VGG16_{bloc3}$	83.23%	94.93%	12.32%	92.33%	68.64%	97.50%	29.60%	**89.94%**
$VGG16_{bloc4}$	83.88%	95.15%	**11.84%**	92.66%	66.26%	97.07%	31.75%	88.81%
$VGG16_{bloc5}$	**84.31%**	96.12%	12.29%	**93.50%**	**80.24%**	92.70%	**13.44%**	89.91%
$VGG16_{bloc2-att}$	74.17%	**98.21%**	24.47%	92.23%	53.54%	**98.99%**	45.91%	84.62%
$VGG16_{bloc3-att}$	**85.45%**	96.57%	**11.50%**	**94.12%**	77.05%	97.36%	20.86%	92.49%
$VGG16_{bloc4-att}$	82.26%	97.41%	15.53%	93.95%	80.36%	97.18%	17.30%	**93.27%**
$VGG16_{bloc5-att}$	84.58%	96.12%	11.98%	93.57%	**81.96%**	96.43%	**15.00%**	93.14%

4.3 Renforcement contre le déséquilibre

Pour finir la présentation des options d'améliorations des réseaux neuronaux, il nous reste le renforcement de l'entraînement des classes sous-représentées et de l'augmentation des données. Ces deux options ont le même objectif : contrebalancer le déséquilibre des données pour diminuer ses effets néfastes sur l'entraînement des réseaux neuronaux. Elles semblent donc toutes indiquées dans notre contexte. Cependant nous devons savoir dans un premier temps si ces méthodes sont bien efficaces dans notre contexte très déséquilibré. Dans un second temps, le questionnement porte sur les effets indésirables que ces méthodes peuvent avoir dans un contexte non déséquilibré. Enfin, est-ce que ces options se combinent bien entre elles et avec les autres options que sont les modules d'attention et la multimodalité ? En somme, il s'agit de savoir les bénéfices que l'on peut espérer de ces méthodes et les comparer au contre-coût que l'on peut estimer.

4.3.1 Fonction de coût pondérée

Afin de renforcer le système contre le déséquilibre, la fonction de coût a été modifiée pour prendre en compte la sous-représentation de certaines classes. La fonction de coût de base utilisée pour entraîner les réseaux de classification est la fonction Cross-Entropy (CE) (plus précisément dans notre cas : "Categorical Cross-Entropy"). Cependant, celle-ci accorde naturellement plus d'importance aux grandes classes qui constituent une part plus importante du coût, dû à la manière dont elle est calculée. De fait, comme le coût des erreurs est calculé pour chaque échantillon, plus une classe est représentée plus la somme de l'ensemble du coût lié à ses échantillons constitue une part importante du coût total calculé sur une étape de l'entraînement. En conséquence, cette classe influe plus le calcul du gradient de correction qu'une autre moins représentée.

Afin de l'adapter au déséquilibre, nous proposons de la modifier pour pondérer la part de coût générée par chaque classe dans le calcul du résultat final. La pondération est opérée par un vecteur de poids N, dont la taille est équivalent au nombre de classes. Le poids assigné à chacune des classes est l'inverse de leur pourcentage de représentation dans le corpus d'entraînement. Par exemple, si les échantillons d'une classe représentent 3% du corpus, alors le poids associé est de 97%. Le calcul de la nouvelle fonction de coût correspond alors à l'équation 4.2.

$$Loss_{CE}(i) = -N_i log(P(i)) \tag{4.2}$$

Cette méthode compense alors directement la sous-représentation d'une classe en augmentant proportionnellement le coût généré par ses échantillons, compensant alors leur plus faible nombre par rapport aux classes plus représentées.

Cette proposition peut avoir des effets négatifs sur les performances du réseau liées aux classes les plus représentées, puisque leurs importances sont diminuées. Elle pourrait également, dans le même ordre d'idées, rendre la convergence du réseau plus difficile (et donc rallonger la durée de l'entraînement). La question que la partie expérimentation doit alors permettre de trancher est : est-ce que le gain que l'on peut attendre de ce module est plus important que ses effets néfastes (si ceux-ci s'avèrent fondés) ?

4.3.2 Augmentation des données

La deuxième méthode utilisée est la génération d'échantillons pour compléter les classes sous-représentées et ainsi réduire le déséquilibre. La génération est réalisée par trois types de transformations simples appliquées à l'image du document. Le premier type d'altération est une rotation aléatoire de quelques degrés (une rotation trop forte étant plus facile à détecter et à corriger en pré-traitement par les outils dont Yooz dispose déjà). Le second est un décalage de l'image dans une direction aléatoire (erreur classique lors de l'utilisation d'un scanner) et le dernier est une suppression d'une partie de l'image, passée au noir (altération due à des dégâts au niveau du document numérique). Ces altérations imitent des cas de documents endommagés probables dans notre contexte. La génération d'échantillons n'est appliquée que à l'image, le texte étant trop difficile et trop risqué à altérer de cette façon. La variété des altérations reste assez limitée (en essayant tout de même de prendre des dégâts fréquemment observés) d'une part car cela s'éloigne un peu du sujet de la thèse et d'autre part car une plus grande variété d'altérations augmente mécaniquement le nombre de documents générés. Ce dernier augmente beaucoup la durée des expérimentations ce qui se traduirait en une réduction du nombre d'expérimentations (que nous avons préféré privilégier).

Quelques exemples de génération sont visibles sur la Figure 4.5. Dans un cadre équilibré (RVL-CDIP au complet), l'augmentation est appliquée sur toutes les classes uniformément (en doublant donc le nombre d'échantillons). Dans les cadres déséquilibrés, elle est utilisée pour doubler seulement les classes sous-représentées, rééquilibrant ainsi la représentation entre les petites et les grandes classes.

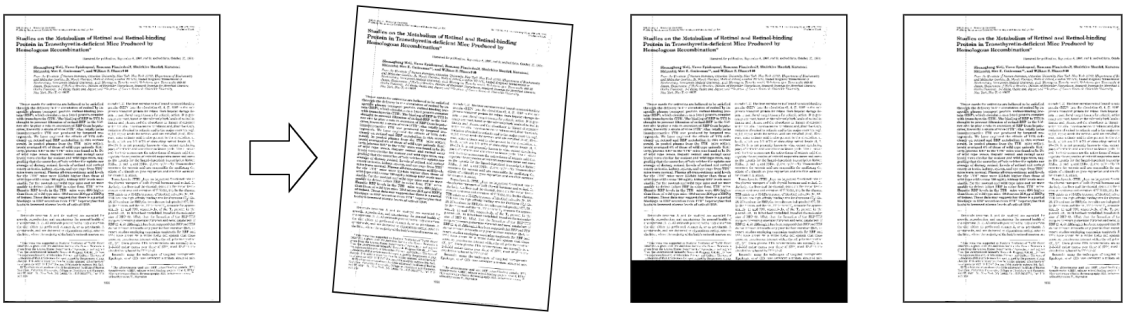

FIGURE 4.5 – Génération d'images avec les trois types d'altérations

4.4 Résultats

Les différentes propositions présentées précédemment ont été testées sur les quatre corpus YoozDB, RVL-CDIP, RVL-Déséquilibré et RVL-Réaliste pour comparer et analyser leurs potentiels dans notre contexte. Elles ont tout d'abord été testées séparément, puis en les combinant pour évaluer autant leurs résultats respectifs que les synergies que ces solutions pourraient avoir entre elles.

L'évaluation sur les corpus YoozDB et RVL-CDIP complet vise principalement à s'assurer que ces propositions n'ont pas trop d'effets néfastes dans un contexte équilibré. Elle sert également à déterminer si certaines des solutions apportent des gains, pour ensuite les comparer avec ceux obtenus dans un contexte déséquilibré. Cela permettrait d'évaluer si les gains acquis sur les corpus équilibrés sont conservés dans un contexte plus proche du nôtre, et dans quelle proportion ils sont conservés.

Comme la plupart des solutions proposées dans ce chapitre vise à améliorer les performances des réseaux dans un contexte déséquilibré, les résultats les plus importants de ces expérimentations sont ceux concernant le corpus RVL-CDIP imitant un déséquilibre. L'efficacité dans un contexte réaliste est également présente à titre indicatif. Il s'agit principalement de voir s'il y a également une amélioration dans ce contexte, bien qu'elle ne soit pas prévue, ainsi que s'assurer que les propositions n'aient pas d'effets néfastes dans ce cas.

4.4.1 Corpus brut et contexte équilibré

Les résultats sur les corpus de YoozDB et de RVL-CDIP complet sont résumés dans le tableau 4.2. Notez que les meilleurs résultats pour une modalité sont en gras noir et ceux toutes modalités confondues sont en gras rouge. Ces résultats dégagent plusieurs éléments concernant l'efficacité des différentes modifications proposées, nous commencerons par l'analyse des modules ajoutés indépendamment. À ce niveau, le module d'attention ($_{att}$) est clairement celui qui est le plus mis en avant par les résultats. Il semble renforcer autant l'accuracy que la précision (à l'exception des cas biRNN/VGG sur la base YoozDB où il y a une perte d'accuracy, mais un gain plus important de précision, selon la F0,5-Mesure). Son effet est plus remarquable sur la base RVL-CDIP que YoozDB, ce qui n'est pas surprenant étant donné que c'est sur la première qu'il y a le plus d'erreurs à rattraper. Ces résultats semblent confirmer nos espérances : l'amélioration de la qualité des caractéristiques dues au module d'attention fonctionne aussi pour les documents, réduisant en conséquence le nombre d'erreurs.

Les options de génération d'images ($_{gen}$) et d'adaptation de la fonction de coût ($_{loss}$) sont un peu plus décevantes. Les résultats sont souvent moins bons que pour la version basique, sauf pour de la génération sur YoozDB qui semble offrir quelques gains (+0,71% de F-0,5M avec VGG16 et +0,13% de F-0,5M avec le multimodale). Cependant, il convient de rappeler que ces modules sont surtout conçus pour des contextes déséquilibrés. Il n'est pas attendu ici qu'ils améliorent les résultats, mais seulement de ne pas trop les diminuer (ce qui est le cas). Dans l'ensemble, tant que les résultats obtenus dans des contextes plus proches de celui de la thèse (soit les corpus déséquilibré et réaliste) sont suffisamment intéressants alors ces options en vaudront la peine. La version combinant tous les modules ($_{all}$) se dispute la meilleure place avec la version utilisant uniquement le module d'attention. Cela semble indiquer que le modèle avec module d'attention tire mieux partie des

autres modules que la version de base.

TABLE 4.2 – Résultats sur les corpus originaux (sans modifications)

	Corpus	YoozDB				RVL-CDIP			
Modalité	Méthodes / Mesures	Accuracy	Précision	T. Rejet	F0,5 M.	Accuracy	Précision	T. Rejet	F0,5 M.
Image	$VGG16$	84.31%	96.12%	12.29%	93.50%	80.24%	92.70%	13.44%	89.91%
	$VGG16_{att}$	82.26%	97.41%	15.53%	93.95%	80.36%	**97.18%**	17.30%	93.28%
	$VGG16_{loss}$	81.15%	95.27%	14.46%	92.07%	76.87%	93.73%	17.99%	89.79%
	$VGG16_{gen}$	81.89%	**97.89%**	16.35%	94.21%	71.57%	95.79%	25.29%	89.72%
	$VGG16_{all}$	**86.19%**	97.05%	**11.20%**	**94.66%**	**86.71%**	96.33%	**9.99%**	**94.24%**
Texte	$biRNN$	**95.60%**	98.25%	**2.70%**	97.71%	75.80%	88.23%	14.09%	85.43%
	$biRNN_{att}$	93.57%	**99.22%**	5.69%	**98.04%**	**77.95%**	88.58%	**12.01%**	86.23%
	$biRNN_{loss}$	92.39%	98.51%	6.22%	97.22%	75.40%	86.76%	13.10%	84.22%
	$biRNN_{all}$	95.06%	98.45%	3.44%	97.75%	74.65%	**91.04%**	18.01%	**87.21%**
Multi	$VGG - biRNN$	96.45%	97.09%	0.66%	96.96%	84.83%	89.65%	5.38%	88.64%
	$VGG - biRNN_{att}$	96.82%	**97.34%**	0.52%	**97.24%**	**89.85%**	**91.34%**	1.63%	**91.04%**
	$VGG - biRNN_{loss}$	93.34%	94.62%	1.21%	94.36%	87.31%	87.32%	**0.02%**	87.32%
	$VGG - biRNN_{gen}$	96.55%	97.23%	0.70%	97.09%	80.94%	89.88%	9.95%	87.94%
	$VGG - biRNN_{all}$	**96.86%**	97.30%	**0.46%**	97.21%	89.79%	90.39%	0.67%	90.27%

Pour finir, le modèle multimodal est un peu décevant. Bien que son accuracy soit fréquemment meilleure que pour les modèles monomodaux, sa précision laisse à désirer. Dans un contexte plus classique où la précision est aussi importante que le reste, ce modèle serait objectivement plus performant que les deux autres. Cependant, la précision a ici plus d'importance et le problème du modèle multimodal provient du taux de confiance qu'il accorde dans ses décisions (qu'elles soient correctes ou erronées).

Nous entendons ici par taux de confiance, la valeur la plus élevée du vecteur one-hot renvoyé par le réseau lors de la classification (soit la certitude avec laquelle il désigne une classe comme étant celle du document d'entrée). Ce taux de confiance est très élevé pour le réseau multimodal avec une moyenne de 0.997 (pour rappel, la valeur est comprise entre 0 et 1). En d'autres termes, dans la très large majorité des cas, la réponse du réseau est un vecteur de 0 avec un 1 pour la classe supposée du document (montrant ainsi, une confiance maximale dans ces décisions). En conséquence, il devient difficile d'appliquer un système de rejet permettant d'écarter les réponses les moins fiables. Ce qui donne ce résultat où la précision est très proche de l'accuracy avec un taux de rejet presque nul.

Cela étant, elle reste la plus équilibrée des architectures sur les deux bases (puisqu'elle est deuxième dans les deux cas). Le tableau 4.3, donne les résultats classe par classe pour RVL-CDIP avec l'architecture basique. Le réseau se montre plus équilibré que la version biRNN (notamment pour les classes "File Folder" et "Handwritten"). Ces résultats sont finalement plus proches de ceux de VGG16 que de ceux de biRNN pour RVL-CDIP, et l'inverse pour YoozDB. Les résultats montrent donc que l'architecture multimodale tire partie des deux modalités. Elle semble s'adapter en fonction du corpus qu'elle doit traiter en privilégiant plus ou moins les caractéristiques d'une des deux modalités en fonction du contexte.

Pour conclure, le module d'attention ($_{att}$) est un ajout de choix dans un contexte équilibré de classification de documents. L'adaptation de la fonction de coût ($_{loss}$) semble avoir un effet néfaste dans un contexte équilibré. Il reste alors à voir les gains qu'elle peut appor-

TABLE 4.3 – Résultats classe par classe de l'architecture multimodale (sans option) sur RVL-CDIP (Les valeurs de rappel n'inclut pas les rejets dans leurs calculs).

Méthode	Multi							
Classes	**Letter**	**Form**	**Email**	**Handwr**	**Advert**	**Sc Report**	**Sc Public.**	**Speci**
Précision	88.62%	83.25%	97.07%	87.95%	85.43%	83.21%	90.93%	95.22%
Rappel	86.56%	80.83%	94.73%	90.62%	89.16%	81.62%	91.87%	93.68%
Taux de Rejet	3.56%	4.04%	0.52%	11.72%	7.40%	5.76%	2.60%	1.32%
F0,5-Mesure	88.20%	82.76%	96.60%	88.47%	86.15%	82.89%	91.12%	94.91%
Classes	**File F**	**News A**	**Budget**	**Invoice**	**Presen**	**Questi**	**Resume**	**Memo**
Précision	82.41%	86.55%	89.47%	93.18%	85.32%	92.63%	98.29%	92.88%
Rappel	93.16%	89.47%	90.06%	91.30%	81.50%	90.69%	97.54%	91.47%
Taux de Rejet	9.36%	2.76%	3.04%	2.96%	6.16%	4.60%	0.72%	2.48%
F0,5-Mesure	84.36%	87.12%	89.58%	92.80%	84.53%	92.24%	98.14%	92.59%

ter dans le contexte déséquilibré, pour lequel elle est conçue. La génération d'images ($_{gen}$) a un effet mitigé, tantôt meilleur, mais souvent moins bon que la version sans module. La combinaison des modules ($_{all}$) a des résultats comparables à la version avec seulement l'attention, bien que dans l'ensemble légèrement moins bon. Le modèle multimodal est la meilleure des architectures si on combine les résultats des deux corpus, mais souffre d'une précision plus faible qui l'empêche de dépasser les deux autres sur les bases qui les avantages (YoozDB pour le biRNN et RVL-CDIP pour VGG16).

4.4.2 Contexte déséquilibré

Les contextes plus proches de celui des flux changent plusieurs choses par rapport aux précédents. Les résultats dans les contextes déséquilibrés et réalistes sont résumés dans le tableau 4.4. Ici aussi, les meilleurs résultats pour une modalité sont en gras noir et ceux toutes modalités confondues sont en gras rouge. Veuillez noter également que pour le corpus réaliste, l'accuracy maximale qui puisse être attendue pour ces réseaux est 75% (étant donné qu'un quart du corpus de test est dédié aux classes incomplètes, représentées seulement par un exemple). Quel que soit le corpus, le module d'attention ($_{att}$) semble toujours être aussi efficace. Le gain de performance acquis dans le contexte équilibré semble pleinement conservé malgré le contexte déséquilibré ou réaliste, voire encore plus efficace selon les architectures. Le module d'attention semble définitivement une amélioration de choix pour les réseaux de classification de documents, y compris dans des contextes difficiles.

Le module de fonction de coût pondérée ($_{loss}$) montre des résultats plus encourageants sur le corpus déséquilibré que sur le corpus réaliste. Il semble principalement améliorer la précision et parfois plus l'accuracy. Cependant, ce gain est plus faible (voire inexistant) avec l'architecture VGG16 qu'avec les deux autres. Pour celui-ci, le module de génération ($_{gen}$) est clairement plus efficace que l'adaptation de la fonction de coût. Au regard de ces résultats, nous recommandons plus d'utiliser la modification de la fonction de coût pour biRNN, le module de génération pour VGG16 et les deux pour l'architecture multimodale. En se basant sur la F0,5 Mesure, la version la plus efficace pour toutes les architectures est celle combinant l'ensemble des modules ($_{all}$). Contrairement au contexte équilibré, les résultats se démarquent plus de ceux de la version avec uniquement le module d'attention. Cela montre qu'il y a bien synergie entre les modules. Comme dans notre cas les contextes

déséquilibré et réaliste sont plus proches de notre problématique que les corpus complets, nous conserverons plutôt la version combinant tous les modules que les autres.

TABLE 4.4 – Résultats aux testes d'adaptations (RVL-CDIP)

	Corpus	Déséquilibré				Réaliste			
Modalité	Méthodes / Mesures	Accuracy	Précision	T. Rejet	F0,5 M.	Accuracy	Précision	T. Rejet	F0,5 M.
Image	$VGG16$	59.17%	88.90%	33.44%	80.78%	52.73%	79.57%	33.63%	75.78%
	$VGG16_{att}$	71.78%	90.65%	**20.82%**	86.12%	54.26%	82.26%	34.04%	78.29%
	$VGG16_{loss}$	55.99%	92.04%	39.17%	81.54%	53.95%	78.91%	31.64%	75.48%
	$VGG16_{gen}$	63.21%	**96.02%**	34.18%	86.99%	**56.51%**	77.97%	**27.53%**	75.17%
	$VGG16_{all}$	**72.24%**	95.19%	24.12%	**89.50%**	56.35%	**86.11%**	34.55%	**81.87%**
Texte	$biRNN$	63.83%	80.47%	20.67%	76.48%	41.15%	59.55%	**30.64%**	57.05%
	$biRNN_{att}$	67.07%	82.36%	18.58%	78.77%	**46.71%**	73.53%	36.07%	69.60%
	$biRNN_{loss}$	**69.71%**	80.59%	**16.76%**	78.15%	39.88%	63.78%	37.42%	60.24%
	$biRNN_{all}$	60.68%	**86.53%**	29.88%	**79.74%**	45.02%	**75.65%**	40.42%	**70.91%**
Multi	$VGG - biRNN$	84.18%	82.55%	1.95%	82.87%	53.71%	66.88%	**19.50%**	65.31%
	$VGG - biRNN_{att}$	83.89%	87.64%	4.28%	86.86%	57.57%	73.86%	22.07%	71.87%
	$VGG - biRNN_{loss}$	84.61%	86.04%	1.67%	85.75%	51.92%	70.41%	26.20%	68.03%
	$VGG - biRNN_{gen}$	**84.90%**	85.40%	**0.58%**	85.30%	53.04%	71.17%	25.47%	68.86%
	$VGG - biRNN_{all}$	83.11%	**88.08%**	5.64%	**87.04%**	**58.39%**	**75.02%**	22.15%	**72.98%**

L'architecture multimodale montre des résultats encourageant sur le corpus déséquilibré, les performances absolues (sans rejet) sont meilleures que celles de VGG16. Cependant, elle souffre du même problème que précédemment : sa précision ne suit pas. La situation est la même que pour le cas équilibré (voir section précédente), les taux de confiance que les réseaux avec cette architecture (toutes options confondues) accordent dans leurs décisions sont trop élevés. En conséquence, elles ne permettent pas d'exploiter correctement le système de rejet, et ce même avec le seuil au maximum. Ce défaut est encore plus problématique dans le cas réaliste, où les réseaux font beaucoup de confusion entre les classes incomplètes et les classes entraînées (classes ayant plus d'un échantillon).

Il ressort de ces expérimentations que les modules d'attention ($_{att}$) sont vivement conseillés dans les deux cas (déséquilibré et réaliste). Toujours dans les deux cas, le module d'adaptation de la fonction de coût ($_{loss}$) est plutôt suggéré pour les architectures multimodales et biRNN. Tandis que pour l'architecture VGG16, c'est le module de génération d'images ($_{gen}$) qui semble être le plus adapté, là où celles multimodales peuvent combiner les deux. La combinaison de l'ensemble des modules est la version qui offre les meilleurs résultats quels que soient le corpus ou l'architecture considérée, il s'agit donc de cette version-là que nous conseillons le plus dans un contexte déséquilibré ou réaliste. Au niveau des architectures, il y a débat entre l'architecture image $VGG16_{all}$ et la multimodale $VGG - biRNN_{all}$ pour le corpus déséquilibré. Cependant pour le contexte réaliste c'est clairement l'architecture VGG16 qui prend l'avantage tant que le problème de précision du modèle multimodal n'est pas réglé.

4.5 Conclusion sur la multimodalité, les adaptations et les systèmes d'attention dans un contexte de flux de documents

Dans cette section nous nous sommes attardés sur l'analyse et l'expérimentation de plusieurs options développées pour tenter de renforcer l'efficacité des réseaux neuronaux dans un contexte de classification de documents déséquilibré. Il en ressort que plusieurs de ces pistes ont du potentiel.

Tout d'abord, l'utilisation de modèles d'attention pour renforcer les architectures des réseaux semble être indispensable, au vu des gains de performances qu'ils offrent sans contreparties. L'utilisation de fonctions de coût pondérées semble être indiquée dans un contexte déséquilibré pour les réseaux textuels, mais à éviter dans un contexte équilibré (puisqu'ils ont tendance à diminuer légèrement les performances dans ce cas-ci). Il en va de même pour la génération d'images. Les résultats obtenus à la suite de nos expérimentations tendent à démontrer qu'une combinaison de ces différents modules est encore plus efficace qu'une utilisation séparée (toujours dans un contexte déséquilibré).

Pour finir, les expérimentations autour de la multimodalité affichent plusieurs avantages attendus. En effet, ce type d'architecture semble permettre de gérer une plus large variété de classes de documents. Cela en fait une architecture moins efficace pour des types particuliers, mais plus polyvalente. Elle est autant capable de classer des corpus plus orientés sur l'image (comme RVL-CDIP) que ceux sur le texte (comme YoozDB), sans trop perdre en efficacité par rapport aux réseaux plus spécialisés. Elle semble également moins souffrir du déséquilibre, en pouvant se reposer sur la modalité la plus discriminante pour apprendre plus efficacement avec moins d'exemples l'ensemble des classes. Cependant, les résultats montrent également un défaut inattendu. Ces architectures ont une précision plus faible que les autres, ce qui est un défaut majeur dans notre cas. Si ce type d'architecture doit être utilisé dans un contexte similaire au nôtre, il faudrait rechercher des solutions à ce problème de précision. Cependant toutes ces propositions n'offrent de solution ni aux problématiques liées à l'incomplétude (les classes non présentes lors de l'entraînement), ni à la classification des très petites classes.

Chapitre 5

Modèle multi-systèmes et cascade

Le réseau développé dans le chapitre précédent permet un apprentissage renforcé contre le déséquilibre, mais il reste le problème des petites et très petites classes. Celles-ci nécessitent des méthodes plus adaptées à une classification en "few-shot", pour apprendre le plus possible d'un minimum d'échantillons. Dans le chapitre 3, les expériences que nous avons mené ont montrées deux candidats potentiels pour résoudre ce problème : ProtoNet et A2ING. Ils sont plus efficaces sur ces petites classes et sont moins perturbés par le manque de données d'entraînement. Cependant, il faudrait trouver comment utiliser ces méthodes pour renforcer des réseaux profonds classiques comme biRNN, VGG et le nouveau modèle multimodal, sans réduire les performances sur les grandes classes. Ce chapitre se concentre sur une méthode de combinaison en cascade permettant de renforcer un réseau pour la classification des petites classes et le déséquilibre avec des systèmes spécialisés. Ces méthodes spécialisées proviennent des domaines du few-shot learning, de l'apprentissage incrémental ou potentiellement d'autres domaines que l'apprentissage (ex : un classifieur de titres à base de règles).

5.1 Préambule

5.1.1 Multi-systèmes

Afin de renforcer les réseaux profonds pour les petites classes, l'option qui sera développée ici est celle d'une combinaison entre un ou plusieurs systèmes de réseaux profonds et des méthodes permettant un entraînement à partir d'un minimum d'exemples. L'idée sous-jacente est de résoudre le problème de la classification des classes les plus difficiles avec une méthode spécialisée (plus efficace), tout en conservant les réseaux profonds développés précédemment pour traiter les autres classes (en essayant de conserver le meilleur des deux). Pour combiner plusieurs méthodes, les stratégies les plus simples à première vue consisteraient à faire une jonction des deux systèmes soit en amont (fusion précoce) soit en aval (fusion tardive) [72].

Une combinaison en amont nécessite du méta-apprentissage pour déterminer avant entraînement quel système se voit attribué quels échantillons du corpus. Il faudrait pour cela un "système d'aiguillage" permettant d'attribuer à chaque système la partie du corpus d'entraînement qui correspond à sa spécialité. Une fois la phase d'entraînement terminée, il faudrait pouvoir combiner au mieux pour chaque exemple les résultats de tous les systèmes pour savoir lequel doit s'exprimer, dans cette situation. Cette forme de combinaison

se base fortement sur le système d'aiguillage qui paraît complexe à concevoir, ce qui en fait probablement la solution la plus difficile à mettre en place.

Une combinaison en aval consisterait à entraîner directement tous les systèmes avec l'ensemble des données, puis à combiner les résultats à posteriori. La décision est emportée par le système "qui parle le plus fort" (le plus confiant dans sa décision). Ces décisions peuvent être pondérées pour favoriser les systèmes les plus précis. Dans l'ensemble ce modèle de combinaison est le plus simple à mettre en place mais également le moins intéressant car il est déjà expérimenté du côté de Yooz. Il est également plus chronophage (si les résultats sont calculés système par système) ou plus coûteux en puissance de calcul cumulée (si les résultats sont calculés simultanément). Ce modèle limite également les possibilités d'améliorations via la complémentarité des différentes méthodes ou d'exploiter de potentielles synergies qu'il pourrait y avoir entre elles, car son champ d'action est limité aux modifications à posteriori.

Il existe néanmoins une dernière possibilité de combinaison qui pourrait mieux convenir que les deux précédentes : faire une cascade. Le concept consiste à utiliser des systèmes successivement. Chaque système s'exprime à tour de rôle sur les documents qu'il peut classer et renvoie ce qu'il n'a pu traiter au suivant. Le point le plus important d'une cascade est l'ordre dans lequel les systèmes sont agencés. La stratégie de cascade déjà testée par Yooz consiste à ordonner du plus au moins précis, en entraînant chaque système (pour les méthodes qui nécessite un apprentissage) sur les données disponibles.

L'idée qui sera développée dans ce chapitre est un peu différente. L'ordre est changé pour devenir : de la méthode la plus générale à la méthode la plus spécifique. Ainsi, les premiers classifieurs sont des méthodes très performantes (réseaux profonds) qui arrivent à classer la majorité du flux (grandes classes) et qui ont besoin de beaucoup de données pour être entraînés. Le premier classifieur s'entraîne sur la totalité du corpus puis transmet au système suivant la partie de ce corpus qu'il n'a pas réussi à apprendre. Le système suivant fait de même et le processus se poursuit jusqu'à la fin de la cascade. La taille du corpus d'entraînement est réduite progressivement avant d'arriver à des méthodes conçues pour du few-shot learning qui peuvent se spécialiser pour traiter les documents les plus difficiles et les classes très peu représentées. L'objectif est d'améliorer la cohérence des systèmes entre eux pour que ceux-ci se complètent au fur et à mesure de la cascade pour couvrir l'ensemble du flux.

5.2 Cascade de systèmes

5.2.1 Principes

L'idée de la cascade de systèmes se fonde pleinement sur une stratégie de diviser les problèmes pour mieux les résoudre, en assignant à chaque problème une méthode qui lui est adaptée. Les systèmes se suivent les uns après les autres en transférant les cas qu'ils n'arrivent pas à résoudre au niveau suivant de la cascade. Pour ce faire, chaque décision n'est acceptée que si son taux de confiance est très élevé. En conséquence, les systèmes se succèdent en étant de plus en plus spécialisés. Les premiers s'occupent de la majorité des cas (qui sont "simples" à résoudre) pour ne renvoyer vers les derniers niveaux que les cas les plus difficiles à trancher pour eux comme les petites classes, les cas particuliers et les isolats. L'ensemble ressemble alors à une succession de tamis aux mailles de plus en

plus fines. La cascade cherche de fait à conserver la plus haute précision possible à chaque étage, avec un rejet à seuil élevé (de toute façon nécessaire dans notre contexte).

Les possibilités de combinaisons offertes par la cascade de systèmes sont nombreuses et l'architecture est pensée pour ne pas être restrictive sur celle interne aux étages, ce qui réduirait le nombre de méthodes utilisables. Par conséquent, il devient possible de combiner les réseaux neuronaux prévus pour servir de première ligne, avec des méthodes qui compensent certaines de leurs faiblesses (ici le déséquilibre et l'incomplétude). Par exemple, cela permet de les associer à des méthodes incrémentales ou spécialisées dans l'apprentissage avec peu d'exemples. Les modalités utilisées par les systèmes peuvent être différentes d'un niveau à l'autre, il est d'ailleurs possible que cela augmente les performances lorsqu'elles sont complémentaires les unes des autres.

5.2.2 Processus d'entraînement

Effectuer une cascade entre plusieurs systèmes pour qu'ils se soutiennent entre eux change la problématique d'origine (créer une méthode qui fasse tout) en une problématique de coordination entre plusieurs systèmes. Si on conserve tel quel le corpus d'entraînement à chaque niveau de la cascade, cela reviendrait finalement à dupliquer les résultats sans gain significatif et à ignorer l'évolution de l'ensemble des données à traiter qui s'opère tout au long de la cascade. Si les systèmes sont tous entraînés avec les mêmes ressources, le niveau suivant (que l'on nommera $n+1$) ne fera que rejeter les mêmes éléments que celui qui le précède (soit n). Or l'objectif de chaque étage est de traiter le plus possible de documents que celui qui le précède a rejeté, changeant ainsi la problématique à chaque niveau de la cascade. En conséquence, pour entraîner efficacement l'étage $n+1$, il est nécessaire de réadapter le corpus d'entraînement à la nouvelle problématique de cet étage : les éléments rejetés par le niveau n. Pour cela, il faudrait que le nouvel ensemble d'entraînement de $n+1$ ressemble le plus possible à ceux que cet étage devra traiter par la suite. Il faut donc retirer de l'ancien ensemble tous les éléments que le niveau n classe sans difficulté et ne garder que les cas qui pourraient être rejetés à l'avenir. La définition du nouveau corpus suit l'équation 5.1, où :

— D_n est le corpus d'entraînement de l'étage n avec $D_n \subset \{d_1, d_2, ..., d_i\}$
— Chaque document d_i a une classe c_j
— D_{n+1} est le corpus d'entraînement de l'étage $n+1$, soit ceux qui ont été rejetés par l'étage n
— D'_n l'ensemble des documents avec un taux de confiance très élevé et dont la classe est suffisamment représentée pour être fiable, d'après l'étage n

$$D_n = D'_n \cup D_{n+1} \tag{5.1}$$

L'évolution du corpus d'entraînement se veut comme une étape permettant de recentrer l'entraînement de l'étage $n+1$, afin de permettre à celui-ci de mieux différencier les documents rejetés par le niveau n, en écartant les cas qu'il ne devrait pas avoir à traiter. Comme pour la cascade les phases d'entraînement sont réalisées successivement, la division du corpus est opérée entre chaque phase pour bien maintenir à jour le corpus tout au long de la cascade, comme illustré par le schéma 5.1.

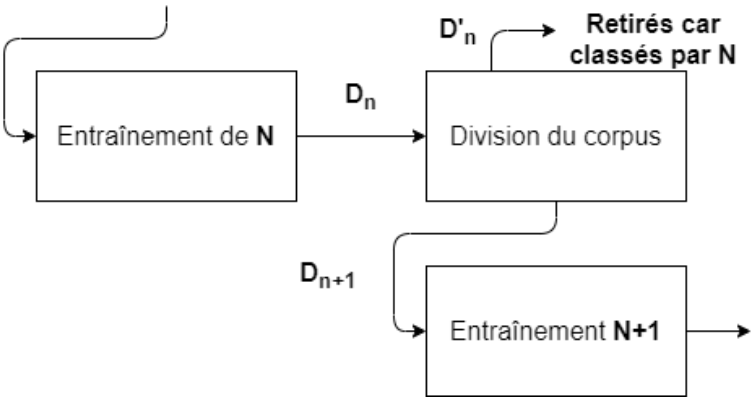

FIGURE 5.1 – Architecture d'entraînement

Avec ce type d'architecture, l'efficacité des étages suivants dépend de la capacité du mécanisme de division à s'approcher d'un D_{n+1} idéal, ce qui en fait alors un point critique de la cascade. D_{n+1} doit contenir autant que possible les éléments proches de ceux qui seront rejetés par l'étage n. La solution la plus simple pour générer D_{n+1} serait d'utiliser uniquement les échantillons que le réseau n'arrive toujours pas à classer à la fin de son entraînement. Le problème de cette idée est que les réseaux neuronaux profonds approchent (voir atteignent) fréquemment les 100% d'accuracy sur le corpus d'entraînement. Il ne reste donc presque aucun document qui remplit cette condition dans la plupart des cas. Il faut alors trouver une méthode permettant de déterminer quels sont les échantillons du corpus D_n les plus difficiles à entraîner pour l'étage n, avec la possibilité de rassembler suffisamment d'exemples pour l'étage suivant.

L'option proposée est d'attribuer un "score de simplicité" à chaque échantillon, dont le calcul est basé sur les informations issues de l'entraînement du réseau (les connaissances à priori étant à éviter autant que possible). Ce score utilise quatre caractéristiques pour évaluer l'efficacité de l'entraînement du modèle de l'étage n sur son corpus d'entraînement. Ces caractéristiques rassemblent des informations provenant du niveau document comme du niveau classe dans l'objectif de décrire au mieux le degré d'intégration de chaque exemple par le modèle de n, et donc s'ils sont susceptibles d'être rejetés ou non. Une fois calculé, le score est utilisé par une fonction présentée dans l'équation 5.2. Cette fonction de sélection compare le score à un seuil pour générer D_{n+1}, permettant de régler la proportion de documents à passer à l'étage suivant.

On définit D_{n+1} comme l'ensemble où $\forall d_i \in D_n$ de classe c_j :

$$d_i \in D_{n+1}, \text{ si et seulement si, } \underbrace{\alpha A}_{\text{niveau document}} + \underbrace{\beta B + \gamma C + \delta D}_{\text{niveau classe}} < T \tag{5.2}$$

Où :

— $\alpha, \beta, \gamma, \delta$ sont des paramètres de pondération, avec $\alpha + \beta + \gamma + \delta = 1$

— A est le taux de confiance calculé par le système de l'étage n (N_n) pour la classe correspondante c_j. Comme les réseaux neuronaux profond proposent généralement des taux de confiance très élevés en toute situation, il est recommandé d'utiliser une fonction de normalisation pour mieux les distinguer entre eux.
Il est préférable d'adapter la fonction de normalisation à la distribution des résultats retournés par le système (par exemple avec une normalisation min/max ou expo-

nentielle) l'objectif étant de ne conserver que les documents avec les scores les plus bas.

— B est l'accuracy obtenue par N_n pour la classe c_j. Cette caractéristique est utilisée pour conserver dans D_{n+1} plus de documents provenant de classe avec une faible accuracy.

— C est le ratio entre la variance inter-classe et intra-classe.
Les variances sont calculées ici avec la distance euclidienne entre les encapsulations des documents E_n entraînées par N_n et leur centroïde de classe. $Ed_n(d_i)$ est la distance entre l'encapsulation de d_i et son centroïde de classe, $Ed_n(c_j)$ correspond à la même chose mais au niveau des classes (soit avec les centroïdes de classes et le centroïde du corpus).
$\overline{Ed_n(c_j)}$ est la distance moyenne pour la classe c_j, de même que $\overline{Ed_n(D_n)}$ avec le corpus. x_j est le nombre d'exemples ayant pour classe c_j et x_n le nombre de classes dans D_n. Comme $C \in [0 : +\infty[$, nous préconisons de borner la valeur maximale, pour conserver l'équité de poids entre les différentes caractéristiques si leurs paramètres de pondération sont tous égaux. Nous avons utilisé pour nos expériences une borne maximale permettant de ramener les valeurs entre $[0 : 2]$. 2 est déjà une valeur très élevée qui suffit pour indiquer qu'il s'agit d'une classe très facile selon ce critère. Elle se traduit par une distance interclasse moyenne deux fois plus supérieure à la distance intraclasse moyenne, assurant qu'elle ne se confond pas avec les autres classes. Ensuite, nous avons normalisé afin de conserver toutes les caractéristiques sur la même échelle $[0 : 1]$ (toujours pour conserver l'équité entre les caractéristiques et faciliter l'utilisation des paramètres pondérations).

$$\begin{cases} C = \dfrac{var_{inter}(c_j)}{var_{intra}(c_j)} \\ \forall c_y \in D_n \\ var_{inter}(c_j) = \dfrac{1}{x_n}\displaystyle\sum_{y=1}^{x_n}(Ed_n(c_y) - \overline{Ed_n(D_n)})^2 \\ \forall d_y \in c_j \\ var_{intra}(c_j) = \dfrac{1}{x_j}\displaystyle\sum_{y=1}^{x_j}(Ed_n(d_y) - \overline{Ed_n(c_j)})^2 \\ \overline{Ed_n(c_j)} = \dfrac{1}{x_j}\displaystyle\sum_{y=1}^{x_j} Ed_n(d_y) \end{cases} \tag{5.3}$$

— D est la représentation de la classe c_j dans D_n avec $size(c_j)$ le nombre de documents dans c_j et $max(size(c))$ le nombre de documents dans la classe la plus représentée dans le corpus. Cette caractéristique assure que les classes les moins représentées restent dans D_{n+1}.

$$D = \frac{size(c_j)}{max(size(c))} \tag{5.4}$$

— T est un paramètre (un seuil) servant à réguler le nombre d'échantillons de D_{n+1} : plus il est haut plus la fonction garde d'échantillons dans D_{n+1}

La sélection est appliquée autant sur le corpus d'entraînement que sur le corpus de validation pour s'assurer que celui-ci reste cohérent pour le niveau $n + 1$. La dernière étape de l'architecture est justement relative à la réduction des corpus d'entraînement et de validation. Dans la majorité des cas, l'équilibre entre le corpus de validation et celui de test est rompu, l'ensemble de validation devenant proportionnellement plus grand. Il faut alors les ajuster en transférant assez d'exemples choisis aléatoirement pour rééquilibrer les ensembles. Cette opération doit être réalisée classe par classe, chacune d'entre elles ayant été réduite différemment. L'architecture globale proposée est résumée par le schéma 5.2.

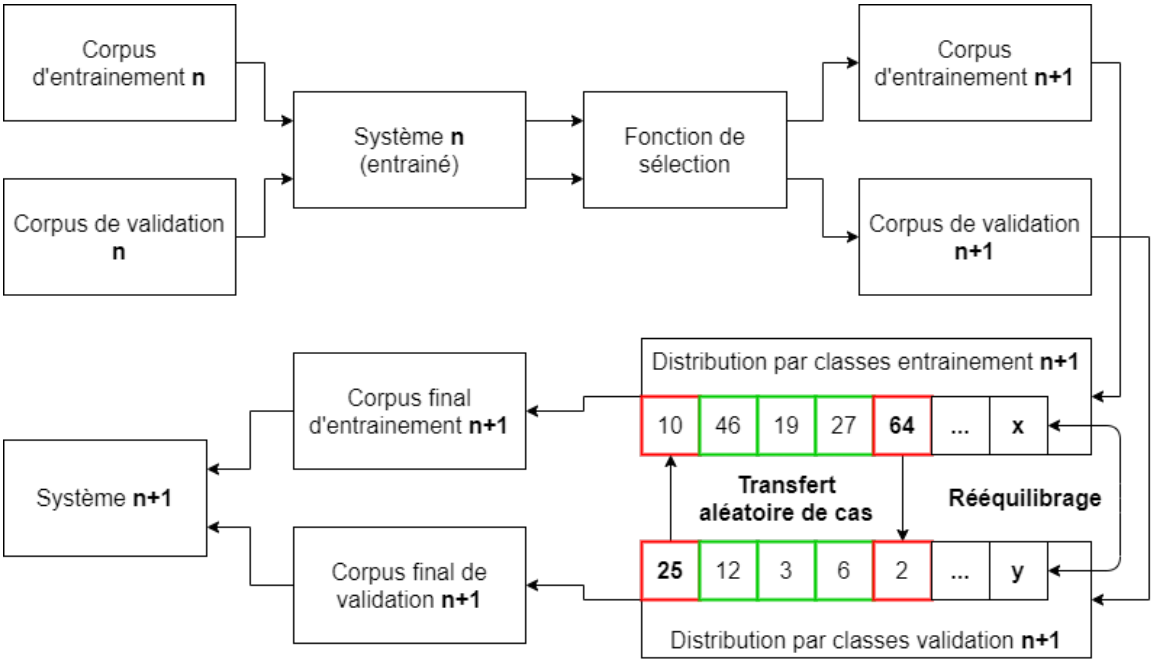

FIGURE 5.2 – Architecture de division du corpus

5.2.3 Processus de décision

Le processus de décision commence une fois que tous les systèmes de la cascade ont terminé leur phase d'entraînement. Ce processus utilise un système de rejet permettant de séparer du reste, les documents dont la classification est la moins fiable pour un niveau n. Ces documents sont ensuite envoyés au niveau $n + 1$ qui tentera de les classer à son tour (et bis repetita). Si aucune décision n'est prise lorsque le document a traversé le dernier étage de la cascade, alors celui-ci est classé en tant que document "rejeté".

Le système de rejet utilise un seuil qui est appliqué sur les taux de confiance renvoyés par le système. Le taux de confiance utilisé dépend de la méthode. Pour les réseaux neuronaux le taux de confiance est la valeur la plus élevée du vecteur de sortie (pour rappel, cela est dû à l'utilisation des vecteurs "one-hot" pour symboliser les classes, voir section 2.1.1). Comme imposé par le contexte, le seuil de rejet est élevé pour tous les systèmes (voir section 1.1).

Pour améliorer ce processus de décision en renforçant encore la coordination entre les étages, un système de pondération est ajouté au processus de décision. L'idée est d'utiliser

le corpus de validation de chaque étage pour estimer une précision à priori du classifieur de cet étage pour chacune des classes. Puis, d'utiliser cette précision à priori pour pondérer le taux de confiance accordé aux décisions portant sur la classe correspondante. Ainsi, le système rejettera plus facilement les documents d'une classe dont la précision est à priori faible ; soit théoriquement les classes qui ont le plus haut risque d'erreur et celles dont le plus d'échantillons ont été envoyés par la fonction de sélection pour l'entraînement de l'étage suivant.

Le nouveau taux de confiance du document d_i (noté $N'_n(d_i)$) est calculé via l'équation 5.5, qui est composée du taux de confiance d'origine et d'une pénalité. Cette pénalité est basée sur la précision de l'étage n pour la classe c_j du corpus de validation (noté $Pre_n(c_j)$). Un paramètre additionnel r est présent pour borner la pondération et faciliter le choix de la valeur du seuil. Ce paramètre évite que le système ne rejette automatiquement des classes entières, mais augmente seulement le taux de confiance exigé par le processus pour valider la décision. Ce système équivaut à une adaptation du seuil de rejet en fonction des classes à partir des informations issues du corpus d'entraînement.

$$N'_n(d_i) = \underbrace{N_n(d_i)}_{\text{Taux de confiance de } d_i} \underbrace{- r(1 - Pre_n(c_j))}_{\text{pénalité pondérée Wb}} \tag{5.5}$$

Le schéma 5.3 illustre l'architecture du processus de décision proposée avec le nouveau système de score pondéré. Ce processus a généralement pour effet de réduire l'accuracy du système de l'étage n, et en contrepartie d'en augmenter la précision. Cet effet touche majoritairement les petites classes, le prochain niveau ayant pour charge de compenser cette perte d'accuracy par une nouvelle classification des rejets, que l'on suppose plus précise.

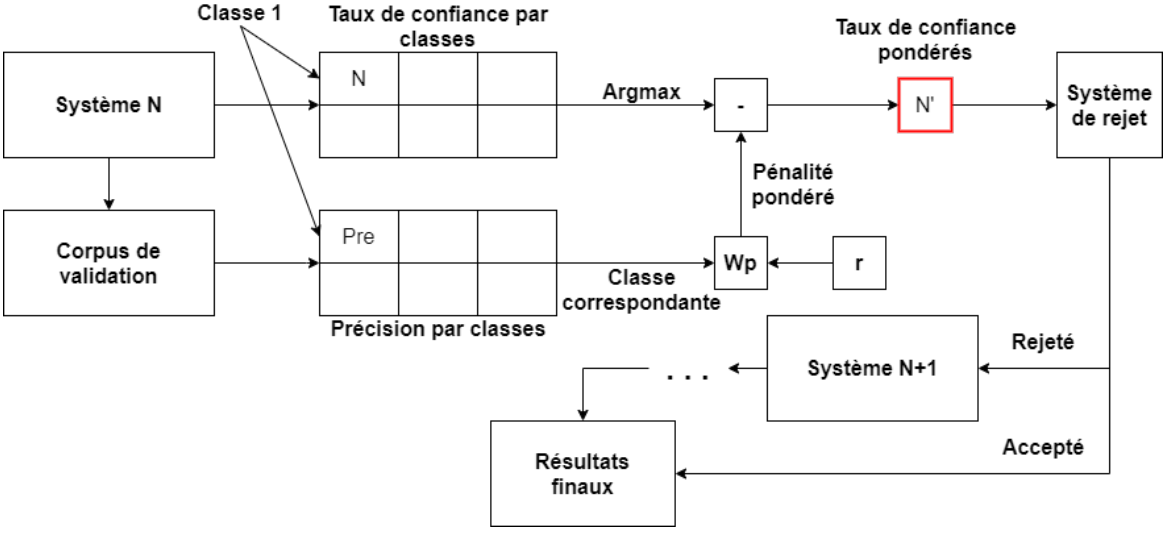

FIGURE 5.3 – Système de pondération par la validation

5.3 Expérimentation

Pour évaluer la méthode de cascade de systèmes précédemment décrite, nous avons utilisé les quatre même corpus que pour le chapitre précédent (soit YoozDB, RVL-CDIP,

RVL-CDIP version déséquilibrée et RVL-CDIP version réaliste, voir les sections 3.1 et 3.2.2). Ici nous n'attendons pas particulièrement d'amélioration sur les deux premiers, le contexte y est équilibré ce qui n'est pas l'objectif de notre nouvelle méthode (même si tout gain est le bienvenu). Ces corpus sont utilisés à titre indicatif, soit pour nous assurer que la cascade conserve des performances au moins équivalentes au premier système dont elle est composée. Les corpus qui nous intéressent le plus sont les deux derniers, puisqu'ils correspondent aux contextes dans lesquels nous souhaitons des améliorations (soit pour rappel les contextes d'entraînement déséquilibré et incomplet, voir section 1.2).

Les réseaux biRNN et VGG16 ont été repris comme composant de base de nos premiers tests en le combinant avec une méthode plus orientée pour l'apprentissage en few-shot ou incrémentale (comme ProtoNet et A2ING). Cela donne une cascade à deux étages visant à utiliser le premier pour traiter la majorité des documents (principalement les grosses classes). Le second est plutôt destiné à traiter les documents les plus complexes à classer, fréquemment associés aux petites classes sur lesquelles ProtoNet a montré être plus efficace que les autres réseaux (voir section 3.3.3). Les combinaisons avec A2ING n'ont été testées que sur la base YoozDB car la RVL-CDIP ne contient pas assez de texte pour cette méthode. Pour prendre en compte les avancées du chapitre 4, le modèle multimodal avec tous les modules a également été expérimenté en combinaison avec le ProtoNet.

Pour finir, nos derniers tests portent sur des architectures en trois étapes. Celles-ci deviennent *de facto* multimodales en combinant les réseaux VGG et biRNN, puis se poursuivent avec un ProtoNet toujours dans l'objectif de mieux traiter les petites classes. Ce type d'architecture a été testé dans les deux sens (avec biRNN ou VGG en premier), afin de pouvoir les comparer. L'idée ici est de tenter une autre forme de multimodalité et de voir si elle peut entrer en concurrence avec l'architecture multimodale de la section 4.1.

5.3.1 Corpus brut et contexte équilibré

Les expérimentations menées sur les corpus YoozDB et RVL-CDIP complet sont résumées dans le tableau 5.1. Les résultats sur la base de Yooz montrent que les méthodes de cascade n'offrent pas de gain significatif dans ce contexte (ce qui était attendu). Ils affichent par contre une perte légère de performance pour les modèles de base les plus efficaces sur ce corpus (Multimodal et biRNN).

C'est surtout le cas pour l'architecture en trois étages biRNN-VGG-ProtoNet, qui n'est pas vraiment adaptée à un corpus aussi petit que YoozDB (le troisième étage est presque inutile car trop peu de données d'entraînement arrive jusqu'à lui). L'architecture en deux étages n'apporte pas grand chose car la perte de précision due à la seconde classification n'est pas compensée par le gain d'accuracy.

Cependant, il est à noter une mention honorable à la combinaison biRNN-A2ING, qui affiche une précision encore meilleure que celle d'origine (cela est dû à la très grande précision de A2ING même avec des documents difficiles à classer). Cette combinaison est d'autant plus intéressante que la méthode A2ING peut se renforcer avec le temps, utilisant un apprentissage incrémental et que cet avantage ne diminue pas beaucoup les performances du réseau. Elle peut également se targuer d'avoir la meilleure précision que nous ayons pu obtenir sur ce corpus, toutes expérimentations confondues.

Les résultats de la combinaison utilisant VGG16 comme base montrent une amélioration des performances par rapport à VGG seul, cependant il reste nettement inférieur aux

autres architectures.

TABLE 5.1 – Résultats sur les corpus brut

Corpus	YoozDB				RVL-CDIP			
Méthodes / Mesures	Accuracy	Précision	T. Rejet	F0,5 M.	Accuracy	Précision	T. Rejet	F0,5 M.
Modalité du premier étage : Image								
ProtoNet	63.56%	**97.93%**	35.10%	88.37%	63.68%	91.36%	30.30%	84.05%
VGG	84.70%	95.47%	11.28%	93.10%	75.14%	**94.41%**	20.41%	89.80%
VGG-ProtoNet	84.21%	95.61%	11.92%	93.09%	75.79%	94.04%	19.41%	89.72%
VGG-A2ING	82.05%	96.62%	15.08%	93.31%	—	—	—	—
VGG-biRNN-ProtoNet	**89.62%**	94.64%	**5.31%**	**93.59%**	**83.22%**	92.06%	**9.60%**	**90.14%**
Modalité du premier étage : Texte								
A2ING	79.07%	97.88%	19.22%	93.43%	22.58%	**95.94%**	76.46%	58.15%
biRNN	93.57%	99.22%	5.69%	**98.04%**	77.95%	88.58%	12.01%	86.23%
biRNN-ProtoNet	**93.77%**	98.95%	5.23%	97.87%	80.72%	88.99%	**9.30%**	87.21%
biRNN-A2ING	91.45%	**99.30%**	7.90%	97.62%	—	—	—	—
biRNN-VGG-ProtoNet	93.75%	97.23%	**3.58%**	96.51%	**81.08%**	90.61%	10.52%	**88.53%**
Modalité du premier étage : Multi								
Multi$_{all}$	**95.53%**	**98.27%**	2.79%	**97.17%**	89.71%	**90.42%**	0.79%	**90.28%**
Multi$_{all}$-ProtoNet	95.51%	97.40%	**1.94%**	97.01%	**89.84%**	90.33%	**0.55%**	90.23%

Sur RVL-CDIP la dynamique est assez différente, puisque les méthodes en cascade se montrent dans l'ensemble légèrement plus performantes que le premier étage seul (notamment pour biRNN). La différence de résultats entre les combinaisons confirme, autant que pour YoozDB, que l'ordre des méthodes est un paramètre important. De plus on peut remarquer que la combinaison utilisant VGG16-biRNN-ProtoNet montre une F0.5-Mesure très proche de l'architecture multimodale, tout en ayant une meilleure précision. Cela est d'autant plus intéressant que la combinaison n'utilise aucun des modules du Chapitre 4, là où Multi$_{all}$ les utilisent tous.

Dans l'ensemble, les architectures en cascade montrent qu'elles n'ont pas d'effet néfaste notable dans un contexte équilibré. Au contraire, elles permettent d'améliorer les performances par une complémentarité des modalités entre les étages de la cascade, au point de rattraper les modèles multimodaux. La combinaison avec A2ING se montre particulièrement intéressante, car elle permet une incrémentalité tout en ayant de meilleures performances que la méthode seule.

5.3.2 Adaptation aux flux de documents

Après l'analyse des résultats dans un contexte équilibré, il est temps de porter plus d'attention à ceux liés aux contextes plus proches de celui des flux de documents. L'ensemble des résultats sur les deux corpus modifiés sont présentés dans le tableau 5.2.

Pour commencer, au niveau des résultats concernant le test d'adaptation au déséquilibre, le modèle Multi$_{all}$ conserve les meilleurs résultats. Sa combinaison avec le Prototypical network ne fonctionne pas bien à cause du manque de précision de l'architecture multimodale. La problématique liée à son taux de confiance trop élevé dans ses décisions

n'aide pas ici (voir la section 4.4.1 pour les détails liés à ce problème du modèle multimodal), car il perturbe le mécanisme de division du corpus d'entraînement. Cela compromet le principe d'adaptation du corpus d'entraînement de notre cascade, visant à rendre le ProtoNet spécialiste dans la classification des rejets du modèle multimodal. Et de fait, le prototypical network échoue à améliorer les performances, parce qu'il y a trop peu de documents rejetés par le premier étage et parce que les documents que le modèle multimodal a du mal à classer sont plus difficiles à identifier (faute de nuance dans ses classifications).

Pour les quatre autres combinaisons, l'architecture en cascade montre une vraie amélioration des performances qui se traduit par une forte augmentation de l'accuracy au dépend de la précision. C'est assez flagrant au niveau des combinaisons avec VGG16, où celle à trois étages où la F0.5-Mesure augmente de presque 5% malgré une perte de plus de 2% de précision. Cette augmentation est due à un gain de plus de 23% d'accuracy (soit pratiquement un quart du corpus de test). La cascade à deux étages est plus précise mais n'augmente l'accuracy que de 11% (moitié moins). Ces résultats portent la version de la cascade à trois étages au même niveau que l'architecture multimodale (et toujours sans les modules de renforcement). Du côté des cascades avec le biRNN comme premier étage, les résultats sont moins flagrants mais tout aussi intéressants puisqu'ils se montrent supérieurs sur tous les points pour la version à trois étages (par rapport à la version sans cascade). Cette augmentation de la précision s'explique par la présence du réseau VGG16 au deuxième étage (qui est un réseau plus précis que le biRNN sur les données de RVL-CDIP) et le fait que le rejet adaptatif selon les classes permet une augmentation de la précision du premier étage surtout sur les classes difficiles de RVL-CDIP (comme "Handwritten" et "File folder"). VGG16 est beaucoup plus précis sur ces classes que ne peut l'être le biRNN.

TABLE 5.2 – Résultats aux tests d'adaptations (RVL-CDIP)

Corpus	Déséquilibré				Réaliste			
Méthodes / Mesures	Accuracy	Précision	T. Rejet	F0,5 M.	Accuracy	Précision	T. Rejet	F0,5 M.
Modalité du premier étage : Image								
ProtoNet	62.55%	**90.60%**	30.96%	83.14%	56.76%	**78.16%**	27.38%	**72.68%**
VGG	59.17%	88.90%	33.44%	80.79%	51.28%	77.97%	34.24%	70.62%
VGG-ProtoNet	70.19%	87.67%	19.94%	83.51%	58.89%	73.28%	19.64%	69.68%
VGG-biRNN-ProtoNet	**82.45%**	86.54%	**4.73%**	**85.69%**	**65.75%**	70.94%	**7.30%**	69.83%
Modalité du premier étage : Texte								
biRNN	68.37%	79.73%	14.25%	77.17%	50.71%	67.56%	24.94%	63.35%
biRNN-ProtoNet	**74.90%**	79.00%	**5.19%**	78.14%	**58.05%**	64.30%	**9.72%**	62.94%
biRNN-VGG-ProtoNet	74.48%	**81.95%**	9.11%	**80.34%**	56.11%	**70.81%**	20.72%	**67.28%**
Modalité du premier étage : Multi								
Multi$_{all}$	82.99%	**86.92%**	4.53%	**86.11%**	59.63%	**76.18%**	21.73%	**72.17%**
Multi$_{all}$-ProtoNet	**83.59%**	86.33%	**3.18%**	85.77%	**63.03%**	73.92%	**14.73%**	71.45%

Pour la partie avec le corpus réaliste, les résultats sont plus décevants, surtout pour les combinaisons utilisant VGG en premier étage. La perte de précision augmente significativement à l'inverse du gain d'accuracy (même si il reste très élevé). Cela rend les résultats beaucoup plus mitigés entre les cascades et le modèle seul, cela se voit à la proximité des F0,5-Mesure. Dans un cas où l'importance de l'accuracy serait équivalente ou supérieure à celle de la précision, le modèle en cascade serait plus performant (surtout celui à

trois étages), mais ce n'est pas le cas ici. Pour les versions avec biRNN en premier étage, l'utilisation d'une cascade améliore les performances. Cependant elles restent toujours inférieures aux autres propositions. Le contexte réaliste semble posé plus de problèmes pour l'entraînement de l'étage final avec le ProtoNet et les problèmes de précision lié à ce contexte. La version avec l'architecture multimodale conserve les mêmes problèmes que précédemment. En somme, aucune des propositions présentes ici ne dépasse la F0.5-Mesure du modèle VGG_{all} (81.87%, voir section 4.4.2)

Pour conclure, les modèles en cascade montrent des résultats très intéressants dans un contexte déséquilibré où ils permettent une augmentation très forte de l'accuracy pour un peu de précision. La version de cascade à trois étages est celle qui apporte le plus, mais également celle dont la précision est la moins bonne. Sur RVL-CDIP, il semble que l'utilisation de VGG16 en premier étage soit préférable à celle du biRNN. La cascade avec le modèle multimodal ne fonctionne pas très bien dû à son trop fort taux de confiance. Si les résultats sont bons sur le corpus déséquilibré, pour un contexte réaliste ce modèle en cascade est beaucoup moins convaincant. Le prototypical network ne se montre clairement pas aussi efficace ici. Il nous faudrait essayer d'autres méthodes de few-shot ou d'apprentissage incrémental ici. L'impossibilité de véritablement évaluer A2ING sur les documents RVL-CDIP est très gênante.

5.3.3 Analyse par classe

Pour approfondir l'analyse des gains de performances offert par les modèles en cascade dans un contexte déséquilibré, nous avons effectué une comparaison classe par classe entre le réseau sans et avec cascade. Les résultats pour la base biRNN sont visibles dans le groupe de tableau 5.3 et ceux de VGG16 sur le tableau 5.4 (à regarder en couleur).

Les résultats attendus sur ces expériences supposent peu de changement sur les groupes de classes 100% et 50% (pourcentage en référence à la proportion de documents d'entraînement conservés pour rapport à RVL-CDIP complet pour la classe). Avec quelques pertes de précision et de rappel pour une réduction légère du taux de rejet (soit un plus grand nombre de classification, mais avec un ratio d'erreur plus élevée). Une amélioration sur les classes 10% et encore plus sur 5% et ce principalement au niveau du taux de rejet et du rappel contre une perte de précision. Avec en tête un élément perturbateur potentiel qu'est l'impact de la multimodalité pour biRNN (puisque le ProtoNet est une méthode utilisant l'image).

Pour la comparaison cascade/sans-cascade avec le biRNN, les résultats sont un peu surprenants. L'impact de la multimodalité est plus fort que nous ne le supposions. Le Prototypical Network est plus adapté (par l'entraînement ciblé de la cascade) pour renforcer les classes complexes pour l'approche texte de l'étage initial que les classes sous-représentées (principalement sur les classes "File Folders" et "Handwritten", ainsi que "Advertisement" et "Presentation" dans une plus moindre mesure). Cet effet rend les gains sur les groupes 10% et 5% plus mitigés que prévu bien qu'ils soient bien présents.

Pour la seconde expérience, avec VGG16 et ProtoNet, les résultats sont plus conformes de ceux initialement attendus. Cela montre bien que précédemment, les perturbations étaient bien dues à l'impact multimodalité. Les gains sur les classes à 10% sont une forte augmentation du rappel contre une quantité nettement moindre mais tout de même important de précision, avec une réduction du taux de rejet autour de 20%. Pour les classes

TABLE 5.3 – Comparaison par classe pour le biRNN et la cascade biRNN-ProtoNet sur RVL-CDIP déséquilibré (Notez que les valeurs de rappel n'inclues pas les rejets dans leurs calculs).

Méthode	biRNN							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	60.90%	62.14%	64.42%	50.82%	86.03%	90.05%	85.07%	95.56%
Rappel	88.02%	82.83%	81.54%	92.06%	88.76%	95.82%	92.79%	98.17%
Taux de Rejet	15.84%	35.00%	35.44%	5.84%	6.76%	2.48%	5.16%	1.72%
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	88.06%	85.60%	85.43%	94.95%	93.81%	93.57%	95.29%	97.26%
Rappel	65.68%	73.93%	54.64%	77.61%	75.54%	57.91%	54.59%	85.07%
Taux de Rejet	16.44%	16.08%	18.88%	10.84%	14.16%	19.60%	15.44%	8.36%

Méthode	biRNN - ProtoNet							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	62.79%	64.91%	73.15%	49.69%	84.45%	89.76%	83.73%	95.52%
Rappel	87.68%	87.36%	87.08%	90.11%	86.56%	95.60%	91.09%	97.42%
Taux de Rejet	4.24%	2.52%	2.16%	3.32%	3.88%	0.96%	3.08%	0.96%
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	86.33%	84.58%	85.77%	93.30%	93.92%	87.54%	88.67%	96.82%
Rappel	62.46%	71.51%	51.65%	74.57%	72.58%	61.90%	58.24%	82.65%
Taux de Rejet	10.60%	10.72%	11.32%	6.56%	10.56%	4.24%	2.20%	5.68%

Méthode	Différence biRNN-ProtoNet - biRNN							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	**1.89%**	**2.77%**	**8.73%**	**-1.13%**	**-1.58%**	-0.29%	**-1.35%**	-0.04%
Rappel	-0.35%	**4.53%**	**5.54%**	**-1.94%**	**-2.20%**	-0.22%	**-1.70%**	-0.75%
Taux de Rejet	**-11.60%**	**-32.48%**	**-33.28%**	**-2.52%**	**-2.88%**	**-1.52%**	**-2.08%**	-0.76%
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	**-1.73%**	**-1.02%**	0.34%	**-1.65%**	0.12%	**-6.03%**	**-6.63%**	-0.43%
Rappel	**-3.22%**	**-2.42%**	**-2.99%**	**-3.04%**	**-2.95%**	**3.99%**	**3.65%**	**-2.42%**
Taux de Rejet	**-5.84%**	**-5.36%**	**-7.56%**	**-4.28%**	**-3.60%**	**-15.36%**	**-13.24%**	**-2.68%**

TABLE 5.4 – Comparaison par classe pour VGG et la cascade VGG-ProtoNet sur RVL-CDIP déséquilibré (Notez que les valeurs de rappel n'inclues pas les rejets dans leurs calculs).

Méthode	VGG							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	85.51%	91.85%	90.57%	59.18%	85.63%	97.81%	85.50%	94.46%
Rappel	97.58%	99.15%	99.09%	92.93%	92.65%	99.33%	94.73%	96.91%
Taux de Rejet	17.52%	15.48%	7.36%	34.36%	24.36%	4.56%	21.08%	10.60%
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	91.35%	94.25%	92.81%	91.86%	96.99%	96.08%	99.35%	98.92%
Rappel	37.31%	85.63%	59.51%	75.93%	86.29%	79.45%	78.39%	84.34%
Taux de Rejet	63.76%	41.00%	59.20%	49.48%	32.88%	49.40%	60.76%	43.28%

Méthode	VGG - ProtoNet							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	85.56%	85.87%	84.76%	59.87%	84.18%	97.57%	85.44%	94.23%
Rappel	93.66%	99.08%	98.40%	85.73%	90.70%	99.05%	92.89%	95.69%
Taux de Rejet	12.92%	8.48%	5.24%	29.36%	17.84%	2.84%	17.88%	7.16%
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	83.20%	92.27%	87.33%	89.84%	95.72%	92.03%	95.84%	97.82%
Rappel	51.46%	84.00%	69.72%	76.65%	85.92%	82.80%	86.95%	85.06%
Taux de Rejet	45.36%	23.24%	36.32%	28.92%	19.88%	20.24%	17.24%	26.12%

Méthode	Différence VGG-ProtoNet - VGG							
Groupes	100%				50%			
Classes	**Advert**	**File F**	**Handwr**	**Sc Report**	**Budget**	**Email**	**Invoice**	**Resume**
Précision	0.06%	**-5.97%**	**-5.80%**	0.69%	**-1.45%**	-0.25%	-0.06%	-0.23%
Rappel	**-3.91%**	-0.07%	-0.70%	**-7.20%**	**-1.95%**	-0.28%	**-1.84%**	**-1.22%**
Taux de Rejet	**-4.60%**	**-7.00%**	**-2.12%**	**-5.00%**	**-6.52%**	**-1.72%**	**-3.20%**	**-3.44%**
Groupes	10%				5%			
Classes	**Form**	**Letter**	**Presen**	**Questi**	**Memo**	**News A**	**Sc Public.**	**Speci**
Précision	**-8.16%**	**-1.98%**	**-5.48%**	**-2.02%**	**-1.27%**	**-4.05%**	**-3.51%**	**-1.10%**
Rappel	**14.16%**	**-1.63%**	**10.21%**	0.72%	-0.37%	**3.35%**	**8.56%**	0.71%
Taux de Rejet	**-18.40%**	**-17.76%**	**-22.88%**	**-20.56%**	**-13.00%**	**-29.16%**	**-43.52%**	**-17.16%**

du groupe 5% c'est encore plus frappant, car les pertes sont plus faibles au niveau de la précision pour une réduction de taux de rejet encore plus élevée (plus de 40% pour la classe "Scientific Publication").

Dans l'ensemble, ces résultats montrent que le modèle en cascade permet une amélioration des performances principalement sur les classes sous-représentées. Il permet également, dans le cas d'une complémentarité des modalités entre les étages, un renforcement sur les classes difficiles pour la modalité initiale. Cela correspond à nos attentes pour la cascade de système dans un contexte déséquilibré.

5.3.4 Conclusion sur les expériences

Pour conclure, la proposition de cascade de systèmes montre une efficacité marquée dans un contexte d'entraînement déséquilibré. Elle permet une forte réduction du nombre de rejets en échange d'une perte de précision acceptable (en se basant la F0.5-Mesure). Elle permet également une autre forme de multimodalité, puisqu'elle permet des gains dans un cas de complémentarité des modalités entre les étages de la cascade. Cependant, les résultats dans le contexte réaliste sont beaucoup plus mitigés par rapport à nos exigences industrielles, puisque le gain d'accuracy ne permet pas de dépasser la perte de précision (toujours en utilisant la F0.5-Mesure pour la comparaison). Il y a également à ajouter les piètres résultats des cascades utilisant le modèle multimodal, qui ne semble tout simplement pas fonctionner avec le Prototypical Network. Nous discuterons des pistes d'améliorations dans le chapitre suivant avec la conclusion de cette thèse.

Chapitre 6

Conclusion

6.1 Retour sur le contexte des flux de documents et l'état de l'art

Dans cette thèse nous avons défini, dans un premier chapitre, une problématique de classification automatique dans le contexte difficile qu'est celui des flux de documents. Il s'agit du contexte industriel où de très nombreux documents doivent être séparés en un large éventail de classes. Cette séparation intervient avant qu'un traitement d'extraction automatique d'informations spécifiques à chaque classe ne soit appliqué sur le document. Les contraintes industrielles imposent pour cette thèse une gestion de documents multilingue, avec un minimum de paramètres, en un minimum de temps et avec un minimum d'erreurs.

Pour cela, nous avons formalisé notre objet d'étude : le flux de documents. Nous le définissons comme une séquence de documents très hétérogènes apparaissant successivement dans le temps. Il est composé d'un grand nombre de classes plus ou moins proches et surtout avec une représentation très inégale. Cette définition nous a permis de mettre en avant deux propriétés contraignantes des flux de documents : le déséquilibre et l'incomplétude. Nous avons formalisé ces deux propriétés comme étant pour la première la forte inégalité de représentation entre les classes au sein du corpus d'entraînement et pour la seconde l'absence d'une partie des classes à l'initialisation. Pour finir, nous avons mis en avant un autre point d'intérêt qu'est l'aspect multimodal des documents et les difficultés de séparer certaines classes selon qu'on utilise l'image ou le texte. Tout ceci fait des flux de documents un contexte de classification très difficile.

Enfin, nous avons discuté deux grands types de solutions présentes dans l'état de l'art : l'utilisation de réseaux neuronaux ou celle de systèmes experts. Les deux ont des avantages et des inconvénients, mais cette thèse cherche avant tout à proposer une alternative à l'existant du côté de Yooz, entreprise en collaboration sur cette thèse. Cet existant concentre beaucoup de systèmes experts et ceux-ci posent un problème inhérent à la méthode de conception de ces systèmes : l'intégration de la nouveauté. Nous avons déjà vu qu'il s'agit d'une part importante des flux de documents. Cette thèse s'est donc concentrée sur les méthodes par apprentissage et notamment les réseaux neuronaux.

Au sein de l'état de l'art nous avons vu que de nombreuses solutions potentielles sont présentes mais aucune qui ne convienne vraiment. Il y a tout d'abord les réseaux neuronaux classiques, qui ne sont pas spécifiquement adaptés au contexte des flux de documents, mais dont les performances des meilleurs modèles sont parmi les plus élevées de

la littérature. Ces réseaux se répartissent pour la plupart entre les deux modalités du documents : l'image et le texte, avec des avantages et des défauts de chaque côté. Nous avons vu que l'option de la multimodalité commence à se démocratiser dans l'état de l'art avec des résultats plutôt convaincants. Ainsi qu'une autre amélioration des réseaux neuronaux : les modèles d'attention.

En plus des réseaux neuronaux classiques nous avons recherché si il n'existait pas déjà des propositions de renforcement contre le déséquilibre, appliquées ou applicables aux réseaux. Nous en avons ressorti deux catégories. D'un côté les solutions visant à renforcer l'apprentissage en prenant le déséquilibre en compte dans l'entraînement, via des fonctions de coût adapté. De l'autre, l'utilisation de système d'augmentation des données, permettant un rééquilibrage artificiel.

Pour tenter de répondre à la problématique des très petites classes et de l'incomplétude nous avons ensuite présenté un axe d'apprentissage avec très peu d'exemples et un axe d'apprentissage incrémental. Pour le premier nous avons trouvé deux options dans l'état de l'art sous la forme des défis du zero-shot et du few-shot learning. Le premier s'est montré inapplicable dans notre situation à cause d'un pré-requis implicite de ces méthodes qui ne peut être satisfait dans notre contexte, celui d'avoir une autre source d'information décrivant les classes manquantes. Le second propose une grande variété de solutions applicables à notre contexte, mais qui montre des performances globalement moins élevées que les réseaux neuronaux classiques sur leur terrain de prédilection.

L'axe d'apprentissage incrémental de l'état de l'art est concentré sur les solutions de neural gaz et des réseaux neuronaux incrémentaux. Si le premier offre un modèle prometteur déjà appliqué au flux de documents, le second contient quelques solutions très récentes et peu convaincantes. Les réseaux neuronaux incrémentaux sont problématiques notamment car les modèles grossissent conséquemment à chaque ajout d'une nouvelle classe (ce qui n'est pas rare avec les flux). Ils apprennent de manière séquencée avec des phases d'apprentissage plus courtes qu'un réentraînement complet, mais plus longues que pour les autres approches incrémentales (tout en ayant une précision inférieure à un réseau classique qui réapprendrait depuis le début). Cela nous a poussé à les écarter des solutions à retenir pour le moment.

A partir de cet état de l'art contenant des propositions variées, nous nous sommes retrouvés avec un problème d'absence dans l'état de l'art de possibilité de comparaison entre ces cinq axes d'approches. Nous avons donc mis en place un protocole d'évaluation et un comparatif entre ces méthodes, sur la thématique des flux de documents. Cette évaluation a été menée sur deux corpus : la base privée de Yooz et la base publique RVL-CDIP. Ces deux corpus ont des défauts mais se complètent sur plusieurs points. Nous avons proposé des méthodes de génération aléatoire de corpus altéré à partir de RVL-CDIP, afin que la base corresponde plus à la forme d'un flux de documents. Ces propositions permettent d'utiliser le large ensemble de test de RVL-CDIP pour comparer les apports des différentes méthodes dans des contextes déséquilibré, incomplet et réaliste (combinaison des deux précédents).

Notre comparatif des différentes méthodes retenues à partir de l'état de l'art nous a permis de mettre en avant plusieurs problématiques. Les réseaux neuronaux sont très performants dans un contexte équilibré mais le sont moins que les méthodes de few-shot learning dans un contexte déséquilibré (et inversement). Cette différence de performance se fait principalement sur les petites classes qui sont négligées par les réseaux neuronaux

lors de l'entraînement de leurs poids. Les réseaux textuels et visuels ont des performances complémentaires entre les différentes classes. Les méthodes utilisant beaucoup le texte sont fortement désavantagées sur le corpus RVL-CDIP (à cause du faible nombre de mots exploitable). Une analyse classe par classe sur des données limitées nous a montré que les effets du déséquilibre n'étaient pas les mêmes selon les classes et selon la modalité utilisée. La difficulté d'une classe pour une modalité donnée la rend encore plus sensible au déséquilibre.

Cette évaluation nous a permis de mettre en avant quatre méthodes comme étant les plus intéressantes de leurs axes respectifs : le réseaux de classification de texte biRNN, la méthode de classification incrémentale de documents A2ING, le réseau de classification d'image VGG16 et le réseau de classification d'image en few-shot learning ProtoNet.

6.2 Conclusion sur les expérimentations, les solutions proposées et les apports scientifiques

Après la mise en place de ce comparatif, nous nous sommes concentrés sur la problématique du déséquilibre. Pour cela, nous avons utilisé plusieurs approches sous la forme de modules de renforcements pouvant être appliqués séparément ou en combinaisons sur des architectures de réseaux neuronaux. Le premier de ces modules est l'intégration d'un modèle d'attention pour renforcer les performances des réseaux. Le second est l'utilisation d'une fonction de coût pondérée conçue pour adapter le gradient au déséquilibre en renforçant l'importance des classes sous-représentées. Elle utilise pour cela la proportion d'échantillons liés à chaque classe au sein du corpus pour favoriser le poids des plus faibles dans le coût total. Le dernier est un générateur d'images altérées pour augmenter les données des classes sous-représentées et rééquilibrer artificiellement le corpus. Nous avons également mis en avant une architecture multimodale utilisant les modèles que nous avons retenus à partir du comparatif.

Toutes ces propositions ont donné des résultats différents. Les modules d'attention se sont montrés particulièrement efficaces quel que soit le contexte, en renforçant nettement et de façon systématique les performances des réseaux sur lesquels ils étaient appliqués. La fonction de coût pondérée s'est montrée efficace contre le déséquilibre pour les modèles textuels (biRNN) et multimodale, mais moins dans le cas réaliste. Par contre, elle s'est montrée inefficace sur le modèle utilisant l'image (VGG16), car elle semble perturber l'entraînement des couches les plus profondes. A contrario, le système de génération de documents altérés a montré son efficacité sur VGG16 et l'architecture multimodale pour le cas déséquilibré, et toujours avec des résultats plus mitigés sur le corpus réaliste (plus proche de la définition d'un flux de documents). L'architecture multimodale s'est montrée performante dans un contexte déséquilibré, mais avec une faiblesse au niveau de la précision. Ce qui lui est beaucoup plus dommageable dans le contexte réaliste où il sous-performe nettement en confondant avec assurance les classes apprises (entraînées sur de multiple documents) et celles qu'il ne peut pas vraiment apprendre (entraînées sur un seul document).

En complément de ces propositions et toujours dans l'objectif d'adapter les réseaux neuronaux au déséquilibre et à l'incomplétude, nous avons proposé un modèle de cascade

de systèmes avec un entraînement sur un corpus spécifique de l'étage suivant permettant de le réadapter aux faiblesses de l'étage précédent. Pour cela nous avons introduit une méthode de division du corpus, afin de n'y conserver que les documents difficiles. Nous avons également proposé une méthode de rejet adaptatif en fonction des classes pour tenter de compenser la perte de précision prévisible due à une nouvelle classification.

Les résultats obtenus sur ces méthodes sont encourageants pour la problématique du déséquilibre, mais décevants pour celui l'incomplétude. Les modèles de cascade en deux étages semblent plus adaptés pour la base de Yooz, là où ceux en trois étages le sont pour RVL-CDIP. Les combinaisons utilisant l'architecture multimodale se sont montrées inopérantes à cause du manque de précision de cette dernière. Cependant, la cascade de systèmes s'est révélée comme une autre option, potentiellement meilleure, pour faire de la multimodalité dans un contexte de flux de documents. La plupart des combinaisons essayées utilise un Prototypical Network (qui pour rappel est une architecture de réseau neuronal spécialisée dans l'apprentissage sur peu d'échantillons), qui ne s'est pas montré comme la meilleure option en tant que dernier étage. Le modèle A2ING (incrémental neural gaz), était en ce sens plus intéressant, mais le corpus RVL-CDIP n'est pas du tout adapté pour tester cette méthode.

En conclusion, nous soutenons que cette thèse a apportée les éléments suivants :
— Une formalisation de la problématique de flux de documents
— Une veille de l'état de l'art autour de ce sujet
— Une proposition de protocole d'évaluation visant à permettre une analyse des améliorations sur ce sujet
— Un comparatif de plusieurs méthodes de l'état de l'art en utilisant le protocole proposé
— La proposition d'une série de modules et d'architectures pouvant être ajoutée à un réseau neuronal pour le renforcer contre le déséquilibre
— Une proposition de cascade de systèmes utilisant des réseaux de neurones et permettant des performances plus élevées dans un contexte déséquilibré

6.3 Axes exploratoires et pistes de réflexion

En guise de poursuite à la réflexion, nous souhaiterions introduire des axes d'exploration pour poursuivre ces travaux. Tout d'abord, il y a la question de la faible précision des modèles multimodaux que nous avons testés. Nous considérons que les architectures multimodales sont une bonne option de base pour la classification des flux de documents si ce défaut parvenait à être corrigé. Sur ce point quelques pistes nous paraissent intéressantes à creuser comme une modification de la fonction de coût ou une réadaptation des paramètres d'entraînement pour limiter ce problème de taux de confiance trop élevés.

Un autre axe exploratoire serait l'approfondissement de la méthode de cascades. Les possibilités que nous voyons seraient l'utilisation d'autres caractéristiques sur lesquelles baser la division du corpus. Comme autre possibilité, nous avons l'intégration dans la cascade de méthodes sans apprentissage ou encore la mise en place d'un système de rejet plus développé et plus adapté à ce contexte de cascade et ainsi améliorer la complémentarité entre les différents étages.

Il y a également le test de plus de combinaisons différentes avec la méthode de cascade, notamment avec l'intégration des modules du chapitre 4. Ces combinaisons pourraient dépasser les performances du modèle multimodal dans le contexte déséquilibré.

Cette version de la multimodalité a pour nous un avantage certain, car elle permet l'utilisation d'architectures de réseaux plus gourmandes en ressources (notamment au niveau du GPU). En effet, les modèles n'ont pas besoin d'être instanciés dans la mémoire de la carte simultanément pendant l'entraînement (ce qui laisse plus de marge de manœuvre). Il pourrait par exemple être considéré d'utiliser une version non tronquée du modèle d'attention de VGG16 [23]. Cependant, celle-ci a le défaut de voir ses performances fluctuer selon l'ordre dans la cascade, ce qui ajoute un paramètre de plus. A noter que la perte de temps provoquée par les multiples classification des systèmes en cascade est limitée par le fait que seule le première étage traite l'ensemble du corpus (et comme il est censé s'occuper de la majorité des documents, les autres systèmes sont *de facto* plus rapides que lui).

Pour finir, il y a les combinaisons incluant le modèle A2ING, qui ne pouvaient être dûment testées sur RVL-CDIP (dû au manque de texte exploitable dans les documents de ce corpus), mais qui restent une option très intéressante. Elle l'est au niveau de la possibilité de profiter de la capacité d'apprentissage incrémental offerte par A2ING qui pourrait être une vraie solution au problème de l'incomplétude. Surtout que les modèles l'incluant ont montré sur la base privée de Yooz, qu'ils parvenaient à égaler les meilleurs résultats.

Bibliographie

[1] Aaron, B., Tamir, D.E., Rishe, N.D., Kandel, A. : Dynamic incremental k-means clustering. In : 2014 International Conference on Computational Science and Computational Intelligence. vol. 1, pp. 308–313. IEEE (2014)

[2] Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H. : State-of-the-art in artificial neural network applications : A survey. Heliyon **4**(11), e00938 (2018). https ://doi.org/https ://doi.org/10.1016/j.heliyon.2018.e00938

[3] Afzal, M.Z., Kölsch, A., Ahmed, S., Liwicki, M. : Cutting the error by half : Investigation of very deep cnn and advanced training strategies for document image classification. In : 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 883–888. IEEE (2017)

[4] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C. : Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1425–1438 (2015)

[5] Asim, M.N., Khan, M.U.G., Malik, M.I., Razzaque, K., Dengel, A., Ahmed, S. : Two stream deep network for document image classification. In : 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1410–1416. IEEE (2019)

[6] Bahdanau, D., Cho, K., Bengio, Y. : Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv :1409.0473 (2014)

[7] Bakkali, S., Ming, Z., Coustaty, M., Rusinol, M. : Visual and textual deep feature fusion for document image classification. In : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 562–563 (2020)

[8] Bengio, Y., Ducharme, R., Vincent, P., Janvin, C. : A neural probabilistic language model. The journal of machine learning research **3**, 1137–1155 (2003)

[9] Bouguelia, M.R. : Classification et apprentissage actif à partir d'un flux de données évolutif en présence d'étiquetage incertain. Ph.D. thesis, Université de Lorraine (2015)

[10] Bouguelia, M.R. : Classification et apprentissage actif à partir d'un flux de données évolutif en présence d'étiquetage incertain. Ph.D. thesis, Université de Lorraine (2015)

[11] Bouguelia, M.R., Belaïd, Y., Belaïd, A. : A stream-based semi-supervised active learning approach for document classification. In : 2013 12th International Conference on Document Analysis and Recognition. pp. 611–615. IEEE (2013)

[12] Changpinyo, S., Chao, W.L., Gong, B., Sha, F. : Synthesized classifiers for zero-shot learning. In : Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5327–5336 (2016)

[13] Chen, Y.h., Lopez-Moreno, I., Sainath, T.N., Visontai, M., Alvarez, R., Parada, C. : Locally-connected and convolutional neural networks for small footprint speaker recognition. In : Sixteenth Annual Conference of the International Speech Communication Association (2015)

[14] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. : Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv :1406.1078 (2014)

[15] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. : Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv :1406.1078 (2014)

[16] d'Andecy, V.P., Joseph, A., Cuenca, J., Ogier, J.M. : Discourse descriptor for document incremental classification comparison with deep learning. In : 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 467–472. IEEE (2019)

[17] d'Andecy, V.P., Joseph, A., Ogier, J.M. : Indus : Incremental document understanding system focus on document classification. In : 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 239–244. IEEE (2018)

[18] Dauphinee, T., Patel, N., Rashidi, M. : Modular multimodal architecture for document classification. arXiv preprint arXiv :1912.04376 (2019)

[19] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. : Bert : Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv :1810.04805 (2018)

[20] Fei-Fei, L., Fergus, R., Perona, P. : One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)

[21] Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T. : Devise : A deep visual-semantic embedding model. In : Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013)

[22] Glorot, X., Bengio, Y. : Understanding the difficulty of training deep feedforward neural networks. In : Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)

[23] Górriz, M., Antony, J., McGuinness, K., Giró-i Nieto, X., O'Connor, N.E. : Assessing knee oa severity with cnn attention-based end-to-end architectures. arXiv preprint arXiv :1908.08856 (2019)

[24] Graves, A., Mohamed, A.r., Hinton, G. : Speech recognition with deep recurrent neural networks. In : 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6645–6649. IEEE (2013)

[25] Graves, A., Wayne, G., Danihelka, I. : Neural turing machines. arXiv preprint arXiv :1410.5401 (2014)

[26] Grosan, C., Abraham, A. : Rule-based expert systems. In : Intelligent systems, pp. 149–185. Springer (2011)

[27] Harley, A.W., Ufkes, A., Derpanis, K.G. : Evaluation of deep convolutional nets for document image classification and retrieval. In : 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 991–995. IEEE (2015)

[28] He, K., Zhang, X., Ren, S., Sun, J. : Deep residual learning for image recognition. In : Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[29] Hochreiter, S., Schmidhuber, J. : Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

[30] Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J. : Stochastic variational inference. Journal of Machine Learning Research **14**(5) (2013)

[31] Hopfield, J.J. : Artificial neural networks. IEEE Circuits and Devices Magazine **4**(5), 3–10 (1988)

[32] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. : Densely connected convolutional networks. In : Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)

[33] Jetley, S., Lord, N.A., Lee, N., Torr, P.H. : Learn to pay attention. arXiv preprint arXiv :1804.02391 (2018)

[34] Johannsen, G., Alty, J.L. : Knowledge engineering for industrial expert systems. Automatica **27**(1), 97–114 (1991)

[35] Johnson, J.M., Khoshgoftaar, T.M. : Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 1–54 (2019)

[36] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T. : Fasttext. zip : Compressing text classification models. arXiv preprint arXiv :1612.03651 (2016)

[37] Ketkar, N. : Stochastic gradient descent. In : Deep learning with Python, pp. 113–132. Springer (2017)

[38] Kim, Y. : Convolutional neural networks for sentence classification. arXiv preprint arXiv :1408.5882 (2014)

[39] Koch, G., Zemel, R., Salakhutdinov, R. : Siamese neural networks for one-shot image recognition. In : ICML deep learning workshop. vol. 2. Lille (2015)

[40] Kochurov, M., Garipov, T., Podoprikhin, D., Molchanov, D., Ashukha, A., Vetrov, D. : Bayesian incremental learning for deep neural networks. arXiv preprint arXiv :1802.07329 (2018)

[41] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N. : Large scale learning of general visual representations for transfer. arXiv preprint arXiv :1912.11370 (2019)

[42] Krizhevsky, A., Sutskever, I., Hinton, G.E. : Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)

[43] Kuhn, H.W. : The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[44] Lai, S., Xu, L., Liu, K., Zhao, J. : Recurrent convolutional neural networks for text classification. In : Twenty-ninth AAAI conference on artificial intelligence (2015)

[45] Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B. : Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015)

[46] Laskov, P., Gehl, C., Krüger, S., Müller, K.R. : Incremental support vector learning : Analysis, implementation and applications. Journal of machine learning research **7**(Sep), 1909–1936 (2006)

[47] Leipert, M., Vogeler, G., Seuret, M., Maier, A., Christlein, V. : The notary in the haystack–countering class imbalance in document processing with cnns. In : International Workshop on Document Analysis Systems. pp. 246–261. Springer (2020)

[48] Leipert, M., Vogeler, G., Seuret, M., Maier, A., Christlein, V. : The notary in the haystack–countering class imbalance in document processing with cnns. In : International Workshop on Document Analysis Systems. pp. 246–261. Springer (2020)

[49] Lin, E., Chen, Q., Qi, X. : Deep reinforcement learning for imbalanced classification. Applied Intelligence pp. 1–15 (2020)

[50] Liu, X., Meng, G., Xiang, S., Pan, C. : Semantic image synthesis via conditional cycle-generative adversarial networks. In : 2018 24th International Conference on Pattern Recognition (ICPR). pp. 988–993. IEEE (2018)

[51] Mani, I., Zhang, I. : knn approach to unbalanced data distributions : a case study involving information extraction. In : Proceedings of workshop on learning from imbalanced datasets. vol. 126. ICML United States (2003)

[52] Marc, R. : Understanding structured documents with a strong layout (2017)

[53] McCloskey, M., Cohen, N.J. : Catastrophic interference in connectionist networks : The sequential learning problem. In : Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)

[54] Mikolov, T., Chen, K., Corrado, G., Dean, J. : Efficient estimation of word representations in vector space. arXiv preprint arXiv :1301.3781 (2013)

[55] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. : Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv :1310.4546 (2013)

[56] Mirończuk, M.M., Protasiewicz, J. : A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications **106**, 36–54 (2018). https ://doi.org/https ://doi.org/10.1016/j.eswa.2018.03.058

[57] Mitchell, T.M., et al. : Machine learning (1997)

[58] Nagy, G., Seth, S.C., Stoddard, S.D. : Document analysis with an expert system. In : Pattern recognition in practice II. pp. 149–159 (1985)

[59] O'Shea, K., Nash, R. : An introduction to convolutional neural networks. arXiv preprint arXiv :1511.08458 (2015)

[60] Palm, R.B., Winther, O., Laws, F. : Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In : 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 406–413. IEEE (2017)

[61] Pennington, J., Socher, R., Manning, C.D. : Glove : Global vectors for word representation. In : Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)

[62] Ren, S., He, K., Girshick, R., Sun, J. : Faster r-cnn : Towards real-time object detection with region proposal networks. arXiv preprint arXiv :1506.01497 (2015)

[63] Rosenfeld, A., Tsotsos, J.K. : Incremental learning through deep adaptation. IEEE transactions on pattern analysis and machine intelligence (2018)

[64] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. : Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)

[65] Sanger, T.D. : Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural networks **2**(6), 459–473 (1989)

[66] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T. : One-shot learning with memory-augmented neural networks. arXiv preprint arXiv :1605.06065 (2016)

[67] Sarwar, S.S., Ankit, A., Roy, K. : Incremental learning in deep convolutional neural networks using partial network sharing. IEEE Access (2019)

[68] Schuster, D., Muthmann, K., Esser, D., Schill, A., Berger, M., Weidling, C., Aliyev, K., Hofmeier, A. : Intellix–end-user trained information extraction for document archiving. In : 2013 12th International Conference on Document Analysis and Recognition. pp. 101–105. IEEE (2013)

[69] Schuster, M., Paliwal, K.K. : Bidirectional recurrent neural networks. IEEE transactions on Signal Processing **45**(11), 2673–2681 (1997)

[70] Simonyan, K., Zisserman, A. : Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv :1409.1556 (2014)

[71] Snell, J., Swersky, K., Zemel, R.S. : Prototypical networks for few-shot learning. arXiv preprint arXiv :1703.05175 (2017)

[72] Soltana, W.B. : Optimisation de stratégies de fusion pour la reconnaissance de visages 3D. Ph.D. thesis, Ecole Centrale de Lyon (2012)

[73] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. : Dropout : a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)

[74] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A. : Inception-v4, inception-resnet and the impact of residual connections on learning. In : Thirty-first AAAI conference on artificial intelligence (2017)

[75] Tan, H. : A brief history and technical review of the expert system research. In : IOP Conference Series : Materials Science and Engineering. vol. 242, p. 012111. IOP Publishing (2017)

[76] Toharudin, T., Pontoh, R.S., Caraka, R.E., Zahroh, S., Lee, Y., Chen, R.C. : Employing long short-term memory and facebook prophet model in air temperature forecasting. Communications in Statistics-Simulation and Computation pp. 1–24 (2020)

[77] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. : Attention is all you need. arXiv preprint arXiv :1706.03762 (2017)

[78] Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M. : Generalizing from a few examples : A survey on few-shot learning. ACM Computing Surveys (CSUR) **53**(3), 1–34 (2020)

[79] Weston, J., Chopra, S., Bordes, A. : Memory networks. arXiv preprint arXiv :1410.3916 (2014)

[80] Wold, S., Esbensen, K., Geladi, P. : Principal component analysis. Chemometrics and intelligent laboratory systems **2**(1-3), 37–52 (1987)

[81] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. : Google's neural machine translation system : Bridging the gap between human and machine translation. arXiv preprint arXiv :1609.08144 (2016)

[82] Xian, Y., Schiele, B., Akata, Z. : Zero-shot learning-the good, the bad and the ugly. In : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4582–4591 (2017)

[83] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M. : Layoutlm : Pre-training of text and layout for document image understanding. In : Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1192–1200 (2020)

[84] Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C. : Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5315–5324 (2017)

[85] Zhang, X., Zhao, J., LeCun, Y. : Character-level convolutional networks for text classification. In : Advances in neural information processing systems. pp. 649–657 (2015)

[86] Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V. : Learning transferable architectures for scalable image recognition. In : Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710 (2018)

Remerciements

J'aimerais remercier chaleureusement les personnes qui m'ont aidé à réaliser cette thèse :

Merci à Mickael Coustaty pour son implication, son soutien et ses précieux conseils.

Merci à Vincent Poulain d'Andecy pour son apport sur le contexte industrielle de la thèse et sa proposition d'intégrer ces travaux en collaboration avec Yooz.

Merci à Aurélie Joseph pour son implication, son soutien, son aide dans les expérimentations et ses précieux conseils.

Merci à Ibrahim Souleiman Mahamoud pour son assistance dans ces recherches et les expérimentations.

Merci à Jean-Marc Ogier pour avoir dirigé cette thèse, ses précieux conseils et son soutien.

Enfin merci à ma famille pour son aide occasionnelle et son soutien dans ces moments pas toujours faciles.

Classification automatique à partir d'un flux de documents

Joris VOERMAN

Les documents administratifs sont aujourd'hui omniprésents dans notre quotidien. Nombreux et diversifiés, ils sont utilisés sous deux formes distinctes : physique ou numérique. La nécessité de passer du physique au numérique selon les situations entraîne des besoins dont le développement de solutions constitue un domaine de recherche actif notamment d'un point de vue industriel. Une fois un document scanné, l'un des premiers éléments à déterminer est le type, la classe ou la catégorie, permettant de faciliter toutes opérations ultérieures. Si la classification automatique est une opération disposant de nombreuses solutions dans l'état de l'art, la classification de documents, le fort déséquilibre au sein des données d'apprentissage et les contraintes industrielles restent trois difficultés majeures. Ce manuscrit se concentre sur la classification automatique par apprentissage de documents à partir de flux industriels en tentant de solutionner ces trois problèmes.
Pour cela, il contient une évaluation de l'adaptation au contexte des méthodes préexistantes ; suivie d'une évaluation des solutions existantes permettant de renforcer les méthodes, ainsi que des combinaisons possibles. Il se termine par la proposition d'une méthode de combinaison de modèles sous la forme de cascade offrant une réponse progressive. Les solutions mises en avant sont d'un côté un réseau multimodal renforcé par un système d'attention assurant la classification d'une grande variété de documents. De l'autre, une cascade de trois réseaux complémentaires : un pour les images, un pour le texte et un pour les classes faiblement représentées. Ces deux options offrent des résultats solides autant dans un contexte idéal que dans un contexte déséquilibré. Dans le premier cas, il équivaut voire dépasse l'état de l'art. Dans le second, ils montrent une augmentation d'environ +6% de F0,5-Mesure par rapport à l'état de l'art.

Mots clés : classification déséquilibrée, documents, réseaux neuronaux, apprentissage profond

Automatic classification of document streams

Administrative documents can be found everywhere today. They are numerous, diverse and can be of two types : physical and numerical. The need to switch between these two forms required the development of new solutions. After document digitization (mainly with a scanner), one of the first problems is to determine the type of the document, which will simplify all future processes. Automatic classification is a complex process that has multiple solutions in the state of the art. Therefore, the document classification, the imbalanced context and industrial constraints will heavily challenge these solutions. This thesis focuses on the automatic classification of document streams with research of solutions to the three major problems previously introduced.
To this end, we first propose an evaluation of existing methods adaptation to document streams context. In addition, this work proposes an evaluation of state-of-the-art solutions to contextual constraints and possible combinations between them. Finally, we propose a new combination method that uses a cascade of systems to offer a gradual solution. The most effective solutions are, at first, a multimodal neural network reinforced by an attention model that is able to classify a great variety of documents. In second, a cascade of three complementary networks with : a one network for text classification, one for image classification and one for low represented classes. These two options provide good results as well in ideal context than in imbalanced context. In the first case, it challenges the state of the art. In the second case, it shows an improvement of +6% F0.5-Measure in comparison to the state of the art.

Keywords : imbalanced document classification, documents, neural network, deep learning

Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau

17042 LA ROCHELLE