Extraction d'information dans des documents manuscrits anciens

Adeline Granet

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE NANTES

THESE DE DOCTORAT DE

L'UNIVERSITE DE NANTES

COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique et applications, section CNU 27*

Par
Adeline GRANET

Extraction d'information dans des documents manuscrits anciens

Thèse présentée et soutenue à Nantes, le 12 décembre 2018
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Frédéric BÉCHET	Professeur, Aix Marseille Université, LIS
Antoine DOUCET	Professeur, Université de La Rochelle, L3I

Composition du Jury :

Président :	Antoine DOUCET	Professeur, Université de La Rochelle, L3I
Examinateurs :	Frédéric BÉCHET	Professeur, Aix Marseille Université, LIS
	Clément CHATELAIN	Maître de Conférences, INSA de Rouen, LITIS
Dir. de thèse :	Emmanuel MORIN	Professeur, Université de Nantes, LS2N
Co-encadrants :	Harold MOUCHÈRE	Maître de Conférences, Université de Nantes, LS2N
	Solen QUINIOU	Maître de Conférences, Université de Nantes, LS2N

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE NANTES

THESE DE DOCTORAT DE

L'UNIVERSITE DE NANTES

COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique et applications, section CNU 27*

Par
Adeline GRANET

Extraction d'information dans des documents manuscrits anciens

Thèse présentée et soutenue à Nantes, le 12 décembre 2018
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Frédéric BÉCHET	Professeur, Aix Marseille Université, LIS
Antoine DOUCET	Professeur, Université de La Rochelle, L3I

Composition du Jury :

Président :	Antoine DOUCET	Professeur, Université de La Rochelle, L3I
Examinateurs :	Frédéric BÉCHET	Professeur, Aix Marseille Université, LIS
	Clément CHATELAIN	Maître de Conférences, INSA de Rouen, LITIS
Dir. de thèse :	Emmanuel MORIN	Professeur, Université de Nantes, LS2N
Co-encadrants :	Harold MOUCHÈRE	Maître de Conférences, Université de Nantes, LS2N
	Solen QUINIOU	Maître de Conférences, Université de Nantes, LS2N

“We never Give Up. . .
涙の日は少しだけ遠く遠く滲む未来
転んだだけ強くなれる”

Kis-My-Ft2

Sommaire

1 Introduction

Sommaire

1.1 Contexte

L'écriture est le moyen que l'être humain a trouvé pour laisser une trace de son passage et marquer son époque. Elle est aussi une forme de partage de connaissances entre celui qui rédige et son lecteur. Cela nous donne un aperçu du quotidien, de la vie à des instants précis de l'histoire. Les siècles passants, les traces écrites nous restent comme des témoignages du temps. Il serait regrettable de ne pas profiter de cette mine de savoirs.

La conservation du patrimoine est une tâche délicate nécessitant la classification et l'organisation des savoirs. Les institutions responsables du patrimoine sont tournées dans une démarche d'humanité numérique depuis quelques années. Elles possèdent une masse de données digitales nouvellement créée et conservée. Cette première étape étant réalisée, il est nécessaire de faire des démarches comme de la

fouille de corpus, de la reconnaissance d'écriture automatique ou de l'indexation pour que ces documents soient exploitables par le plus grand nombre.

Durant ces vingt dernières années, ces différentes méthodes ont permis d'aider les administrations à automatiser et accélérer les traitements d'informations manuscrites : l'analyse automatique des chèques, la reconnaissance automatique des adresses postales, la classification des demandes administratives pour adresser directement la requête au bon service. Une particularité de ces documents est la redondance des informations dans des zones peu variantes, ce qui simplifie la localisation des informations à extraire. De plus, le vocabulaire est fermé c'est-à-dire restreint par exemple aux chiffres, aux noms de rues, de villes par exemple. Dans le cadre d'études de documents avec un vocabulaire plus large, il est préférable d'allier les compétences techniques avancées en analyse d'images avec le savoir linguistique.

Dans un contexte particulier, les systèmes automatiques sont construits pour répondre une tâche donnée c'est-à-dire ce que l'on souhaite que le système résolve, pour cela un apprentissage doit être effectué à partir d'exemples pour ensuite généraliser la connaissance. Le principal intérêt des systèmes automatiques est qu'ils tendent vers une généralisation des informations et ainsi s'adaptent sur de nouvelles données. En reconnaissance d'écriture manuscrite, ce que l'on définit ici comme un exemple est une association entre une forme (plus particulièrement en reconnaissance d'écriture) et une étiquette qui peut être un caractère ou un mot. Cela suggère d'avoir un ensemble de données, préalablement annotées, appelé "vérité terrain" pour réaliser l'apprentissage, mais également évaluer le modèle, c'est-à-dire mesurer la qualité de l'apprentissage. Une fois les étapes d'apprentissage et d'évaluation passées, il est possible d'appliquer le modèle sur des données encore jamais vues. Le problème majeur que posent ces solutions vient de la quantité de données existantes nécessaire pour réaliser la phase d'entraînement des systèmes. Cependant, réaliser cet apprentissage lorsque l'on travaille sur de nouvelles ressources constituées de milliers de pages, avec une très grande quantité d'information, peut vite s'avérer très couteux. Malheureusement, il n'est pas toujours possible de réaliser cet apprentissage lorsque l'on travaille sur de nouvelles ressources constituées de milliers de pages, avec une très grande quantité d'information.

À chaque étude de nouveaux documents, il faut se poser la question de comment réaliser l'extraction de l'information sans avoir de vérité terrain. Il est courant que les chercheurs prennent la décision d'annoter manuellement une partie des documents pour faciliter ensuite l'utilisation d'un système. Cette thèse s'inscrit dans l'environnement d'un projet de l'Agence Nationale de la Recherche (ANR) du nom de CIRESFI (présenté dans la section 1.2) en collaboration avec la Bibliothèque nationale de France (BnF) autour du théâtre italien du XVIII^{e} siècle.

1.2 Projet CIRESFI

Il est important d'expliquer le contexte politique et culturel du XVII^{e} siècle pour mieux comprendre ce qui s'est passé au cours du siècle suivant ainsi que la motivation du projet. Louis XIV était surnommé le patron des arts, sa politique culturelle l'a conduit à fonder en 1669 l'Académie Royale de musique (Opéra) et en 1680 la Comédie-Française. Il donne à chacun un privilège, à l'Opéra, l'exclusivité des représentations de musique et danse, et à la Comédie-Française, l'exclusivité des spectacles de comédie et tragédie.

Cependant, un autre théâtre, la Comédie-Italienne, est toléré, car les acteurs sont italiens et jouent en italien. Louis XIV les apprécie et leur verse une pension. Peu à peu, ils introduisent du français, attirent un large public qui suscite la convoitise et la jalousie de la Comédie-Française. Les complots de celle-ci conduisent à la fermeture de la Comédie-Italienne de 1697 à 1716.

D'autres représentations théâtrales ont lieu par intermittence lors de la Foire Saint-Germain (février-mars) et de la Foire Saint-Laurent (août-septembre). Dans chacun de ces espaces marchands se développent trois ou quatre théâtres de 1670 environ jusqu'à la fin du XVIII^{e} siècle. Mais ces troupes attirent de plus en plus de monde, tandis que la Comédie-Française perd de plus en plus de public. Elle va donc utiliser son privilège pour multiplier les interdictions et les procès. Les acteurs forains vont faire preuve d'ingéniosité pour continuer à se produire, et vont révolutionner le théâtre de l'époque avec de nouveaux genres de pièces : monologues à plusieurs, pantomimes, pièces par écriteaux (où le public chante), opéras pour marionnettes, opéra-comique.

Malheureusement, cette partie de l'histoire reste méconnue du grand public pour des raisons politiques, idéologiques et matérielles. Mais, de cette époque, une masse de documents manuscrits a été précieusement conservée. Le développement de systèmes interactifs et semi-automatiques d'exploitation des archives de cette époque devrait permettre de faire des progrès sur la compréhension de ces problématiques.

Pour la première fois, la Comédie-Italienne et les théâtres forains, ces deux exclus du système des privilèges, sont traités ensemble afin d'analyser les processus d'acculturation des Italiens et d'institutionnalisation, ainsi que les innovations artistiques inventées sous la contrainte par ces théâtres pour se maintenir. Devant l'ampleur des archives à disposition, le projet CIRESFI [1] souhaite relever un défi

1. Contrainte et Intégration : pour une Réévaluation des Spectacles Forains et Italiens sous l'Ancien Régime. Ce projet ANR a été initié et est dirigé par Françoise Rubellin, professeur à l'Université de Nantes, et spécialiste de Marivaux et des théâtres de la Foire au XVIII^{e} siècle, de la Comédie-Italienne, de l'opéra-comique, de la parodie et des pièces pour marionnettes.

technologique en aidant les spécialistes du théâtre. Pour cela, une méthodologie d'analyse automatique des images de documents est mise en place, ainsi qu'un outil de consultation et d'exploitation de milliers de pages des registres manuscrits couvrant une grande partie du XVIII^{e} siècle.

Le projet se donne pour mission l'analyse de l'adaptation des Italiens qui les a conduits à leur implantation au cœur du siècle des Lumières ; et des innovations artistiques, fruit des persécutions incessantes dont ils ont été victimes. Il se veut audacieux en faisant appel à différents genres : dramaturgique, historique, sociologique et linguistique.

Contextualisation historico-politique Le premier objectif du projet CIRESFI est d'analyser le développement du théâtre italien dans le décor français de l'époque pour comprendre comment il a réussi à obtenir le privilège de l'Opéra sous le nom d'Opéra-comique. Cette analyse doit reconstruire les étapes qui ont mené les Italiens et les forains à s'intégrer et former une institution reconnue à travers les formes concurrentes privilégiées du théâtre de la Foire. Mais également, en étudiant les lieux de diffusion des pièces en dehors de la capitale ; et en mesurant leur intégration dans les grands théâtres français.

Contrainte comme moteur Le second objectif est une étude avancée des pièces de théâtre afin de montrer les évolutions dans les formes dues au contournement des interdictions et à l'évolution des goûts pour le théâtre de la Foire tout particulièrement. L'édition et l'étude des pièces de la Comédie-Italienne doivent témoigner du multilinguisme de l'époque.

Économie des spectacles Le troisième objectif s'attachera à retracer les coûts de production et d'organisation dans les théâtres forains et italiens à travers les acteurs employés, l'apparition des droits d'auteur, et les supports matériels (les décors, les costumes, les accessoires, les instruments de musique, les partitions, les danses). Pour finalement mettre en lumière le caractère social du théâtre.

Exploration et analyse innovante Les registres sont protégés et conservés pour assurer leur pérennisation. La seule manière de les consulter (et exploiter)

Différents partenaires de tous horizons : Le laboratoire l'AMo, "Postérités de l'Antique, Généalogies du Moderne", centre de recherche en littérature de l'Université de Nantes ; anciennement le LINA, Laboratoire d'Informatique de Nantes Atlantique avec l'équipe DUKe (devenu LS2N) ; anciennement l'IRCCyN, Institut de Recherche en Communications et Cybernétique de Nantes équipe IVC (devenu LS2N équipe IPI) ; le CHEC, Centre d'Histoire "Espaces & Cultures" rassemblant des enseignants-chercheurs et chercheurs en histoire, histoire de l'art et archéologie de l'Université de Clermont ; le CERHIC, Centre d'Études et de Recherche en Histoire Culturelle de l'université Reims Champagne-Ardenne ; l'ELCI, Équipe littérature et culture italiennes de Paris-Sorbonne ; et Cambridge university.

sans risque se fait grâce à la numérisation. Cette étape passée, l'extraction d'information doit toujours se faire manuellement. Cela reste un processus long et sujet aux erreurs de lecture, car il faut être capable de faire le tri parmi toutes les informations.

À titre indicatif, la transcription de 10 saisons (10 ans) de registres, par une seule personne, en ciblant uniquement les salaires des auteurs (présent dans les comptes mensuels) nécessite 3 mois. C'est pour faciliter cette recherche fastidieuse et chronophage que la reconnaissance d'écriture et l'analyse des données sont nécessaires à ce projet.

1.3 Registres comptables de la Comédie-Italienne

Parmi les documents jusque là inexploités fournissant des informations autour du théâtre italien de cette période, nous trouvons des actes notariés, des procès-verbaux, des comédies et des opéras-comiques inédits. C'est plus de 30 000 pages de ressources inédites qui sont disponibles. La ressource la plus volumineuse autour de la Comédie-Italienne est un ensemble de registres financiers répertoriant les comptes de la troupe durant leurs années d'exercice.

Le projet CIRESFI a permis la numérisation de 63 registres qui sont représentés sur la frise chronologique sur la figure 1.1. Chaque registre en notre possession est représenté par un rectangle bleu. Les rectangles grisés sont les registres absents des archives de la BnF. L'ensemble des registres de la Comédie-Italienne couvre la période de 1716 à 1791 qui correspond à l'autorisation de revenir jouer à Paris que les comédiens italiens ont eue et à l'abolition des privilèges.

Le corpus est constitué de 27 544 pages au total, mais seulement 25 250 pages sont disponibles en haute qualité. Nous avons identifié plusieurs types de pages : les couvertures, des pages blanches, l'état des pensionnaires (c'est-à-dire les acteurs et actrices faisant partie de la troupe pour la saison en cours), l'ouverture de la saison, les comptes quotidiens, mensuels et annuels. Nous nous focalisons sur les comptes quotidiens représentant entre 30 et 90 % des pages suivant les registres.

La figure 1.2 est un exemplaire type de compte quotidien avec les principales informations : en haut, la date du jour suivi des titres des pièces qui ont été jouées ce soir-là ; les recettes détaillées dans la colonne de gauche et les dépenses à droite ; suivi des noms des actrices et des acteurs du soir ; le tout complété par des notes. Ces documents ne sont pas uniquement intéressants pour comprendre l'économie des théâtres, mais également pour comprendre le contexte et les évènements liés à cette époque.

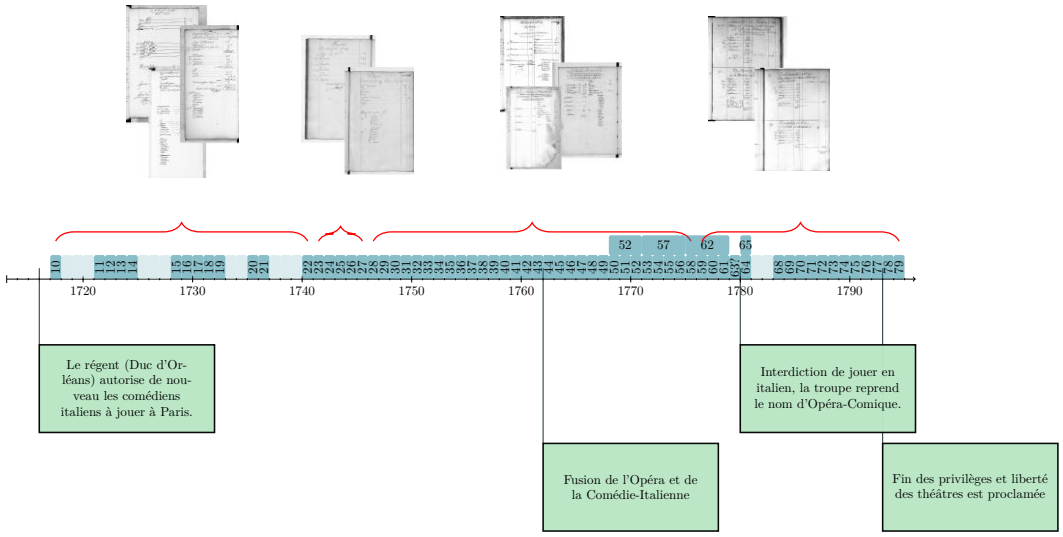


FIGURE 1.1 – Frise historique répertoriant l'ensemble des registres de la Comédie-Italienne.

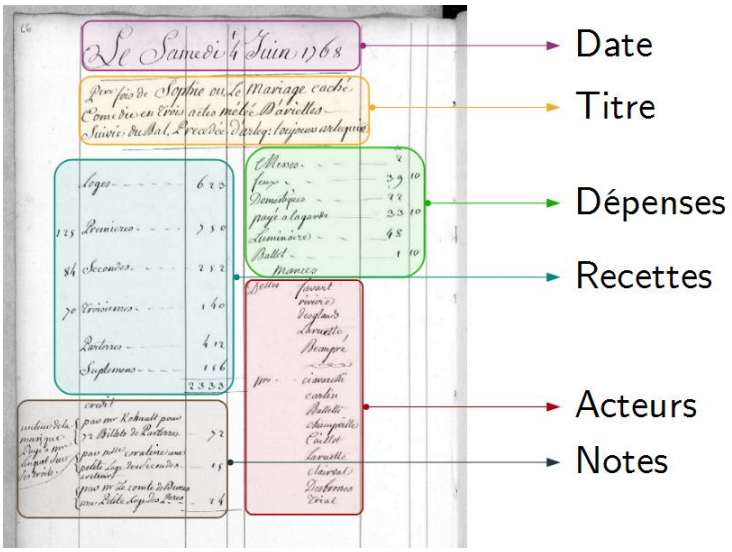


FIGURE 1.2 – Exemple de compte quotidien de la Comédie-Italienne avec une identification des informations types.

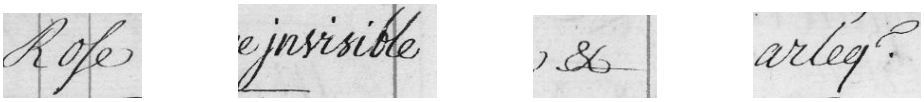
Suite à une étude de ces pages, nous avons pu noter une évolution au fil des registres et des saisons, dans la présentation et la disposition des informations sur les comptes quotidiens.

Nous sommes en mesure de classer en 4 groupes distincts ces pages comme le montre la frise de la figure 1.1. Pendant 12 saisons au cours de 23 premières années, les informations étaient sommaires sur une unique page. Puis, la quantité d'information a évolué nécessitant deux pages (une pour les recettes, et la seconde pour les dépenses), avant d'être réunie sur une unique page. Finalement, les derniers registres reviendront à un format plus concis permettant de rédiger deux jours sur une page.

Une autre évolution notable concerne la langue de rédaction. Durant les premières années, les acteurs italiens rédigeaient eux-mêmes ces registres ce qui fait que les documents sont écrits en italien. L'année 1741 marque l'arrivée de Jean-Baptiste Linguet, le caissier de la Comédie-Italienne embauché pour rédiger les comptes et en français. Cela explique les évolutions notées, qui vont des différents dialectes italiens de l'époque vers le français, ainsi que les évolutions dans le style de l'écriture.

Pour finir, nous avons pu noter des caractéristiques typographiques typiques de cette époque comme :

— la forme longue du caractère "s" qui se retrouve dans d'autres ressources de cette période (Figure 1.3a) ;
— la non-différentiation des caractères "i" et "j" (Figure 1.3b) malgré une réforme au début siècle ;
— un ensemble d'abréviations, comme "&c." pour "etc" pour raccourcir les titres longs, ou l'utilisation de divers symboles dédiés à abréger certains mots (Figures 1.3c et 1.3d) ;
— la substitution du "t" par le "s" pour les mots au pluriel comme "divertissement" qui devient "divertissemens" ;
— les apparitions des accents au milieu du siècle.



(a) "Rose" (b) "Invisible" (c) "etc" (d) "arlequin"

FIGURE 1.3 – Exemples de caractères spéciaux et abréviations dans les documents de la Comédie-Italienne

Des règles orthographiques ont commencé à apparaître au début du XVIII^{e} siècle, mais ont mis du temps à être appliquées.

Pour la suite de notre étude, nous nous sommes concentrés sur la zone de titre uniquement. Cependant, nous ne possédons pas d'informations sur la position de cette zone dans les différentes pages. Il faudra donc être capable de détecter, puis segmenter cette zone.

1.4 Problématiques et contribution

Nous devons faire face à une problématique à deux niveaux. La première porte sur les registres de la Comédie-Italienne eux-mêmes. En effet, ils montrent des spécificités physiques et langagières qui en font un défi technique pour des systèmes de détection automatique de zones, des systèmes de segmentation en lignes ou en mots ou encore des systèmes de reconnaissance d'écriture. Voici quelques-unes de ces caractéristiques :

— des types de zones identifiés, mais non fixes,
— des lignes écrites non rectilignes,
— des hampes et des jambages qui se recoupent,
— des multi-scripteurs,
— des caractères spéciaux liés au siècle,
— des abréviations,
— des données multilingues,
— vocabulaire fermé (les titres des pièces).

Notre objectif est d'automatiser l'analyse de ces documents pour les chercheurs en Sciences Humaines et Sociales (SHS), rendre les informations contenues dans ces registres accessibles et ainsi accroître la connaissance de cette période théâtrale. Malheureusement, la seule chose que nous avons est l'expertise des chercheurs SHS. La seconde problématique est celle-ci. Nous devons mettre en place un système capable d'extraire les informations des registres automatiquement. Or, tous les systèmes connus de l'état-de-l'art requièrent une étape d'apprentissage, et idéalement avec des données provenant de la même ressource pour spécialiser le système.

Au regard du large éventail de systèmes de reconnaissance d'écriture existants et performants, nous ne souhaitons pas créer un nouveau système encore plus complexe et spécifique. Comme nous l'avons déjà évoqué, nous sommes dans une ère où les projets pluridisciplinaires sont en plein essor. Nous souhaitons nous diriger vers une solution qui minimiserait le temps nécessaire pour spécialiser un système à une ressource. L'objectif principal de cette thèse est de mettre en œuvre un système de reconnaissance d'écriture manuscrite capable d'exploiter au mieux les dernières méthodes réalisées en traitement du langage.

La contribution majeure de cette thèse est un système de reconnaissance d'écriture de type encodeur-décodeur permettant un apprentissage des deux composants indépendamment l'un de l'autre : l'un, pour encoder les images de mots et l'autre pour décoder vers le texte. L'atout de cette solution réside dans cet apprentissage indépendant des deux systèmes reliés par un vecteur qui sert de pivot entre les

deux composants représentant les n-grammes de l'image.

L'apprentissage transductif par transfert de connaissances (Pan et Yang 2010a) est une approche intéressante dans le cas où il y a un manque, voir une absence de données pour réaliser l'apprentissage d'un système. En effet, cette méthode consiste à utiliser différentes sources de données pour l'apprentissage d'un système dédié à une tâche, et à l'appliquer sur des données différentes (Pan et Yang 2010b). Il est donc possible pour nous à partir de différentes données connues d'annoter des données provenant d'une source différente. Ce procédé est utilisé pour alimenter les systèmes d'apprentissage gourmands dans différents domaines tels que la détection de mot automatique dans les documents historiques (Lladós et al. 2012) ou pour les modèles multimodaux de traduction (Nakayama et Nishida 2017). Notre idée directrice est d'utiliser l'apprentissage par transfert de connaissances pour faire de la reconnaissance d'écriture sur ces nouveaux documents.

Un second objectif est l'apprentissage par transfert de connaissances sur les registres de la Comédie-Italienne. Nous avons choisi un système simple construit à partir d'un réseau neuronal bidirectionnel avec des cellules *Long Short Term Memory* (BLSTM). Un réseau entièrement constitué de convolution (FCN) est placé en amont de ce système pour extraire les caractéristiques des images. Nous proposons une étude des différents paramètres que nous avons fait varier afin d'optimiser le réseau et ainsi obtenir la meilleure configuration possible pour expérimenter l'apprentissage par transfert de connaissances.

Un dernier objectif lié à ce système est de mettre en avant de nouvelles ressources historiques linguistiques disponibles. Dans les différents domaines du traitement du langage, il est courant de pallier un manque de données en utilisant Wikipédia. Cette ressource présente des avantages certains comme sa taille, son multilinguisme et son large vocabulaire. Cependant, l'orthographe et les mots utilisés sont modernes ce qui peut être un inconvénient pour des travaux sur des données historiques. Nous proposons donc de montrer l'avantage d'aller chercher des ressources sous-exploitées.

Le chapitre suivant présente un état de l'art des systèmes de reconnaissance d'écriture manuscrite hors-lignes existants. Nous survolons les premières étapes de pré-traitement et d'extraction des caractéristiques. Nous avons fait ce choix, car nous ne réalisons pas de pré-traitements sur nos données. Il a été établi par le passé que cela permettait de gommer les différences entre les styles d'écritures et les types de ressources. Cependant cette diversité est une force qu'il faut exploiter, et donc conserver toutes les aspérités. Par le passé, des études ont largement couvert les différentes méthodes d'extraction de caractéristiques en les identifiant parfaitement. Nous avons privilégié une extraction automatique pour les différents systèmes mis en place. C'est pour l'ensemble de ces raisons que nous avons choisi de ne pas

approfondir ces deux points. Le chapitre se concentre sur les principales méthodes de conversion de l'image en séquence de caractères suivant les deux grandes approches connues qui sont les approches stochastiques et neuronales profondes. Pour conclure, nous mettrons en lumière les modèles de langages qui soutiennent les systèmes de reconnaissance ainsi que les méthodes d'évaluations utilisées.

Le troisième chapitre introduit l'ensemble des ressources sur lesquelles nous nous sommes basées pour éprouver nos systèmes pour l'apprentissage par transfert de connaissances. Dans un premier temps, nous présentons la plateforme de crowdsourcing, à travers son fonctionnement, grâce à laquelle nous avons pu constituer une base d'images de lignes sur les registres de la Comédie-Italienne. Dans un second temps, nous décrivons les ressources d'images existantes sélectionnées en insistant sur les caractéristiques de chacune qui nous ont semblé pertinentes pour réaliser notre approche d'apprentissage par transfert de connaissances. Pour finir, nous présentons les ressources sous-exploitées, mais disponibles avec une orthographe et un vocabulaire proche des registres de la Comédie-Italienne, que nous avons été cherchés pour enrichir nos données.

Le quatrième chapitre détaille le premier système que nous avons mis en place pour réaliser l'apprentissage par transfert de connaissances. Nous l'avons construit en nous basant sur les systèmes "simples" de l'état de l'art à partir de réseau bidirectionnel (BLSTM) et d'une fonction de coût facilitant l'alignement entre l'entrée et la sortie du système (CTC). Les résultats obtenus avec ce système ont été publiés dans la conférence ICPRAM en janvier 2018.

Le cinquième et dernier chapitre présente le second système réalisé en combinant deux sous-systèmes complémentaires, mais indépendants. La première partie aborde le concept global en se focalisant sur la première particularité de notre approche : le vecteur pivot qui fait la jonction entre les deux systèmes et projette les informations des images dans un espace non-latent. La seconde partie approfondit la construction du premier composant qui est un modèle optique qui extrait les caractéristiques de l'image et les projette dans un vecteur de n-grammes. La troisième partie se focalise sur le second composant qui décode le vecteur en séquence de caractères. L'apprentissage des deux composants étant indépendant l'un de l'autre, nous proposons des expérimentations dans chaque partie pour pouvoir définir les meilleures configurations de chaque système. Pour finir, les résultats obtenus sur l'ensemble du système avec les deux composants mis bout à bout sont présentés. L'utilisation des composants indépendants et leurs résultats respectifs ont été publiés dans des conférences nationales (TALN 2018) et internationales (COLING 2018, ICFHR 2018).

2

État de l'art en reconnaissance d'écriture manuscrite

Sommaire

Cette thèse se focalise sur la reconnaissance d'écriture manuscrite sur des documents anciens qui n'ont pas encore été transcrits (sans vérité terrain disponible). Pour pouvoir atteindre une transcription automatique de ces documents, quelques étapes doivent être réalisées préalablement. Chacune d'elles est mentionnée sur la figure 2.1. Les premières étapes du processus sont celles de la détection et de la segmentation des zones et de pré-traitements des documents. Parmi les pré-traitements qui peuvent être opérés sur les images, un ensemble d'opérations consistent à nettoyer les images, par exemple en supprimant le fond, et à les normaliser afin de diminuer les différences entre les images. Ces étapes sont grisées sur le schéma, car elles n'ont pas été utilisées dans ces travaux. Cependant, elles peuvent influer sur la suite du processus donc nécessitent une courte présentation. Ensuite, un ensemble de caractéristiques est extrait pour représenter les images, soit de façon ad-hoc soit automatiquement à l'aide de réseaux à convolution (CNN). Elles alimentent, par la suite, le modèle de reconnaissance défini avec l'utilisation de ressources linguistiques pour faciliter la transcription potentiellement. Afin d'obtenir la transcription finale, une dernière étape de décodage ou de post-traitement est réalisée.

Dans ce chapitre, nous présentons les différents types de modèles pour les systèmes reconnaissance d'écriture manuscrite existants. Nous commençons par présenter dans la section 2.1.2, les étapes préliminaires qui peuvent être appliquées sur les documents anciens ainsi que les différentes méthodes d'extractions de caractéristiques existantes. Ensuite, la section 2.2.1 est dédiée aux modèles de Markov (HMM) qui ont dominé les systèmes état de l'art pendant longtemps, et encore aujourd'hui des systèmes hybrides les utilisent encore. Nous nous attardons plus longuement dans la section 2.2.2 sur les réseaux de neurones récurrents (RNN) qui dominent les compétitions dédiées à la reconnaissance d'écriture manuscrite, en détaillant chaque type de modèle avec une mémoire à long terme (LSTM), bidirectionnel, multi-dimensionnel ou intégrant des convolutions (CRNN). L'apprentissage ainsi que les méthodes de décodage des séquences sont détaillés avec la présentation du CTC (*Connectionist Temporal Classification*). Il n'est pas rare pour ces systèmes de faire appel à des modèles de langages pour améliorer les performances, la section 2.3, leur est dédié. La section 2.4 présente la méthode d'apprentissage par transfert de connaissances qui est utilisée lorsque l'on doit faire face à un manque, voire une absence de données d'apprentissage.

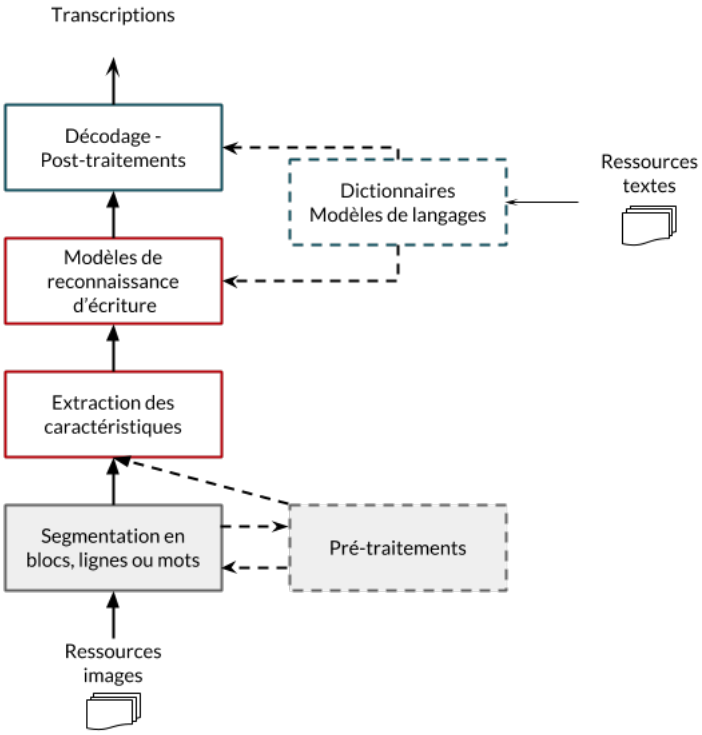


FIGURE 2.1 – Processus global d'un système de reconnaissance d'écriture manuscrite détaillé. Les parties en pointillés sont des étapes optionnelles. Les parties grisées sont des étapes non-réalisées directement dans cette thèse.

2.1 Introduction

2.1.1 Pré-traitements sur les documents anciens

La reconnaissance d'écriture manuscrite est une tâche relativement difficile sur des documents anciens dû aux dommages causés par le temps ou la conservation, au style de l'écriture ou encore à la grande quantité d'information. Suivant la nature du document et l'objectif de l'étude, les pré-traitements sont nécessaires et doivent être adaptés.

Suppression du bruit c'est-à-dire supprimer le fond de l'image qui peut freiner un système de reconnaissance d'écriture manuscrite, car il contient par exemple le verso d'une page, ou des trous et des taches dans les pages, des lignes ou autres ornements pour structurer les informations. Pour cela, un ensemble de filtres (passe-haut, passe-bas ou encore morphologique) peuvent être appliqués afin de différencier et supprimer les éléments superflus, un aperçu de ces approches est fourni par (Ketata et Khemakhem 2010) ; des méthodes (comme Particle Swarm Optimization, PSO) (Quraishi et al. 2013) combinant des filtres bilatéraux et des algorithmes ou par variation totale qui construit une image intermédiaire utilisée comme masque afin d'atténuer le fond avant d'utiliser un filtrage moyen non local (Likforman-Sulem et al. 2011).

Binarisation signifie convertir les images en niveaux de gris en noir et blanc. Le but est de séparer en deux classes bien distinctes le fond de l'encre. Les méthodes les plus simples sont l'utilisation de filtres gaussiens et la définition d'un seuil d'intensité global, appelée Otsu (Otsu 1975). Ces méthodes ne donnent pas toujours l'effet escompté sur des documents dégradés localement, des alternatives locales ont été proposées ((Gatos et al. 2006), (Gatos et al. 2008), (Shi et Govindaraju 2004) ou encore (Su et al. 2010)).

Correction des lignes de texte c'est-à-dire les méthodes qui corrigent l'inclinaison des mots, et des lignes (*skew*). Il n'est pas rare de trouver des lignes incurvées dans les documents historiques, il est préférable de corriger les lignes de bases pour homogénéiser l'ensemble des documents, et favoriser l'extraction de caractéristiques. Il existe différentes méthodes pour corriger les lignes de bases : par projection horizontale des profils (Vinciarelli et Luettin 2001), par interpolation des contours (Bozinovic et Srihari 1989), par estimation de la ligne de base au niveau mot (Lemaitre et al. 2009) ou au niveau de la ligne (Lemaitre et al. 2011 ; Boukharouba 2017) ou encore par correction locale (Espana-Boquera et al. 2011). Il existe plusieurs articles faisant une comparaison de l'ensemble des méthodes

existantes comme (Rehman et Saba 2011).

Correction de l'inclinaison de l'écriture (*slant*) l'angle de l'inclinaison de l'écriture va être modifié pour effacer les différences entre les scripteurs. Pour cela, une estimation de l'angle est effectuée globalement ou localement (Caesar et al. 1993 ; Vinciarelli et Luettin 1999 ; Uchida et al. 2001 ; Zeeuw et al. 2006 ; Bertolami et al. 2007) puis l'image du texte est modifiée, généralement par cisaillement.

Normalisation de la hauteur de l'écriture c'est-à-dire que les zones de hampes et de jambages et la zone centrale sont fixées à une certaine hauteur (Marti et Bunke 2001).

2.1.2 Extraction de caractéristiques

L'étape d'extraction des caractéristiques sur les images pour représenter les informations est primordiale pour assurer un système de reconnaissance d'écriture manuscrite de qualité. Les images de mots, lignes ou blocs qui sont en deux dimensions, sont transformés en un vecteur de caractéristiques ce qui correspond à un ensemble de valeurs numériques représentant un segment de l'image. Ce sont ces informations qui alimentent les systèmes de reconnaissance d'écriture manuscrite. Le calcul des caractéristiques peut se faire par l'intermédiaire d'une fenêtre glissante (à largeur fixe, et hauteur fixe ou adaptée), par segmentation en caractères (ou graphèmes) ou bien sur l'image complète de la séquence. Dans certaines études, les auteurs ont choisi d'utiliser directement les valeurs brutes des pixels de l'image comme caractéristiques d'entrée (Swaileh et al. 2017a). Le choix de la fenêtre glissante est important pour représenter un segment au mieux, mais les pré-traitements sont également importants et vont influer sur les méthodes utilisées. Nous proposons dans cette partie, une liste non-exhaustive des principales méthodes existantes. Nous les classons selon trois catégories : statistiques et structurelles, directionnelles, et par apprentissage.

Caractéristiques statistiques et structurelles L'ensemble de ces méthodes sont au plus proche des vraies valeurs de pixels. Elles utilisent essentiellement des images binaires et la taille de la fenêtre d'observation reste assez restreinte. La longueur du vecteur de caractéristiques obtenues est directement liée à cette fenêtre. Les caractéristiques statistiques se basent sur la valeur directe des pixels alors que les caractéristiques structurelles étudient l'agencement des pixels les uns par rapport aux autres. Ces dernières sont très sensibles au bruit dans les images, il faut donc être prudent quant au choix des pré-traitements. Voici les méthodes les plus utilisées dans cette catégorie :

— la valeur des pixels ;
— le nombre de transitions observées entre l'écriture et l'arrière-plan ;
— le nombre de pixels d'encre observés ;
— la position des contours supérieurs et inférieurs dans la fenêtre ;
— la moyenne des valeurs des pixels ;
— le moment des pixels du contour ;
— la position du centre de gravité ;
— la position des lignes bases (ou références).

Des études répertorient plus largement ces types de méthodes comme (Mohamad et al. 2015).

Caractéristiques statistiques et directionnelles Dans cette catégorie, nous présentons les deux méthodes les plus utilisées et performantes qui permettent de décrire l'orientation des traits par la construction d'histogrammes orientés. La première méthode est une transformation de caractéristiques invariante à l'échelle, appelée *SIFT* (Lowe 1999a ; Lowe 2004). Elle se base sur un ensemble de points clés détectés dans l'image par une convolution à l'aide d'une fonction gaussienne. Puis, un calcul du gradient est opéré pour chacun des points clés, avant de construire l'histogramme des orientations sur 360°. La seconde méthode est l'histogramme des gradients orientés, appelée *HOG*. Cette méthode a été proposée pour de la reconnaissance de gestes (Freeman et Roth 1995) et la détection d'objets (Dalal et Triggs 2005). Depuis, elle a été régulièrement utilisée dans la reconnaissance d'écriture (Terasawa et Tanaka 2009 ; Howe et al. 2009). Comme pour la méthode précédente, l'histogramme des orientations du gradient est construit sur l'ensemble des points de l'image cette fois, ou en la divisant en cellule.

La figure 2.2a montre une image extraite de notre corpus de la Comédie-Italienne qui a été binarisée à l'aide de la méthode de seuillage Otsu, puis sur laquelle a été appliquée la méthode HOG. La figure 2.2b montre le résultat obtenu pour des blocs constitués de 8×8 cellules, chacune composée de 8×8 pixels. La concaténation des histogrammes donne la possibilité d'adapter le niveau d'observation des objets dans une image selon le but de l'étude.

Pour sélectionner les caractéristiques les plus appropriées pour une tâche donnée, des études réalisent une analyse par composantes principales des vecteurs de caractéristiques obtenus (ACP) ce qui permet également de réduire la quantité d'information décrivant une image (Vinciarelli et Bengio 2002).

Caractéristiques par apprentissage Ces dernières années, de nouvelles méthodes d'extraction de caractéristiques par apprentissage se sont répandues dans les

(a) Image origine

(b) Histogramme de gradients orientés

FIGURE 2.2 – Exemple d'application de la méthode HOG avec 8 orientations, cellules constituées de 8x8 pixels, blocs constitués de 8x8 cellules.

domaines du *Vision Object* et en Reconnaissance de la parole. Principalement construites à partir de réseaux de neurones, elles ont l'avantage de pouvoir opérer une extraction non supervisée des caractéristiques sans connaissances à priori sur les images. De plus, il est possible d'obtenir une représentation de l'image à différentes échelles. Il existe différents types de systèmes : des réseaux de neurones simples, à base de convolution ou des auto-encodeurs.

Depuis 2012, la résolution de la tâche de classification d'images a été régulièrement gagnée par des systèmes utilisant ces réseaux à convolution comme (KRIZHEVSKY et al. 2012 ; SIMONYAN et ZISSERMAN 2014a ; HE et al. 2016 ; KÖLSCH et al. 2017), ou dans la détection d'élément dans des images naturels (WANG et al. 2017). Dans le cadre de l'analyse de documents manuscrits, nous retrouvons l'utilisation de réseaux à convolution dans différentes tâches comme la segmentation des documents (HE et al. 2017), la détection de zones (YI et al. 2017), la recherche et classification d'éléments (TENSMEYER et al. 2017) ainsi qu'en reconnaissance de l'écriture comme opérateur de pré-traitement (WIGINGTON et al. 2017 ; TENSMEYER et MARTINEZ 2017) ou simple extracteur de caractéristiques (BLUCHE et MESSINA 2017 ; DING et al. 2017). Les réseaux à convolution sont des réseaux de type *feed-forward* dont la première couche au moins est une convolution. Cette couche de convolution prend en son entrée, des objets en deux ou trois dimensions (comme les images en couleurs par exemple). Elle est définie par un ensemble de filtres de taille restreinte (classiquement ces filtres sont limités à 3 ou 5 pixels de largeur et hauteur) qui s'étendent sur la profondeur. Donc chaque filtre parcourt l'ensemble de l'image afin de calculer le produit scalaire des pixels de la fenêtre. Le résultat est, ce que l'on nomme, une carte d'activation en deux dimensions. Deux hyper-paramètres sont à définir pour contrôler la taille de la carte d'activation :

— le chevauchement pour la fenêtre d'observation, avec une valeur de 1, le filtre se déplace de pixel en pixel sur l'image ;

— la création d'une marge autour de l'image (*padding*) : remplie de la valeur 0, appelée *zero-padding*, ou de la valeur identique à l'image ; si la taille de la marge est égale à la moitié de la taille du filtre, cela permet d'avoir une carte de caractéristiques de la même taille que celle de l'entrée, appelée *same-padding*.

Chaque filtre définit un ensemble de caractéristiques à une échelle différente selon sa position dans l'architecture du réseau. L'avantage de la couche de convolution réside dans le partage des poids par tous les neurones d'une même couche ce qui réduit le temps d'apprentissage. Cette couche est souvent suivie d'une couche de sous-échantillonnage, appelée *Pooling* qui en partitionnant l'entrée, réduit l'image d'entrée. Il faut définir la taille du filtre ainsi que le pas pour appliquer ce dernier. Cette méthode permet de réduire encore un peu plus le nombre de calculs et d'éviter le sur-apprentissage puisque les informations sont réduites. Finalement, (Zheng et al. 2016) présentent les avantages à utiliser les réseaux à convolution comme extracteur de caractéristiques ainsi que les pratiques privilégiées pour améliorer les performances des réseaux.

De nouvelles techniques utilisant les auto-encodeurs à variation (Pu et al. 2016), à convolution (Masci et al. 2011), pour débruiter (Du et al. 2017 ; Rifai et al. 2011 ; Vincent et al. 2010) sont venues enrichir les méthodes non-supervisées existantes d'extraction de caractéristiques discriminantes. Les auto-encodeurs sont des réseaux de neurones composés d'un encodeur qui prend une image x en entrée et active des couches de taille de plus en plus petite pour en obtenir une représentation réduite, et d'un décodeur qui à partir de cette représentation reconstruit une image x' en sortie. Durant l'apprentissage, la différence entre l'entrée et la sortie est calculée pour être ensuite rétropropagée et corriger le modèle. Une fois l'apprentissage terminé, la partie décodeur est retirée du modèle afin d'utiliser la sortie de l'encodeur comme ensemble de caractéristiques. Cette méthode est utilisée pour des tâches telles que le regroupement non-supervisé de documents (Seuret et al. 2015).

2.2 Systèmes état de l'art

Parmi les méthodologies de reconnaissance de l'écriture manuscrite, on peut faire ressortir trois types de systèmes. À la fin des années 90, les modèles de Markov cachés (HMM) se sont imposés en surclassant par leur capacité d'apprentissage sur des séquences les approches structurelles qui prévalaient alors (Bunke et al. 1995 ; Park et Lee 1996 ; Fine et al. 1998). Ensuite, le pouvoir discriminant des réseaux de neurones a permis, à travers des systèmes hybrides neuro-markoviens, de mieux modéliser le caractère local et global de l'écriture (Koerich et al. 2002 ; Fischer

et al. 2012). Enfin, depuis quelques années, les architectures de type BLSTM, définies ci-dessous, intègrent encore mieux cette capacité à mixer du local et du global, avec des effets de contexte, pour optimiser une décision sur une séquence complète (Fischer et al. 2009a ; Grosicki et El Abed 2009a).

2.2.1 Approches stochastiques

Un modèle de Markov caché (HMM) est un modèle probabiliste qui modélise et reconnait des séquences temporelles. Les HMMs sont des processus doublement stochastiques émettant à la fois des probabilités pour passer d'un état vers un autre ainsi que des probabilités d'observations. Le modèle prend en entrée une séquence d'observation, notée $x = x_0, x_1, x_2, \ldots, x_n$, qui correspond au vecteur de caractéristiques extrait souvent à l'aide d'une fenêtre glissante. Pour cette séquence d'entrée x, le modèle cherche à maximiser la probabilité d'observation d'une séquence émise par les états cachés. La probabilité d'émettre une observation $O = (o_0, o_1, o_2, \ldots, o_T)$ représente la possibilité qu'un état caché $Q = (q_1, q_2, \ldots, q_T)$ génère cette observation à un instant t. L'ensemble des probabilités d'émettre une observation pour chaque état est une matrice de taille N×M où les valeurs N et M correspondent respectivement aux nombres d'états et aux nombres d'observations. À chaque instant t, la probabilité d'émettre un symbole k dans un état q_j s'écrit $\mathrm{b}_j(\mathrm{k}) = \mathbb{P}(o_t = v_k | q_t = s_j)$ dans le cadre d'un modèle discret et $\mathrm{b}_j(\mathrm{O}_t) = \sum_{j=1}^{M} \mathrm{C}_{jm}\, \mathcal{N}(O_t, \mu_m, U_{jm})$ dans le cadre d'un modèle continu mettant en œuvre une distribution gaussienne. Les différentes notations entre modèle continu et discret sont regroupées dans la table 2.1.

Un modèle HMM se définit par $\lambda = \{\mathrm{A}, \mathrm{B}, \pi\}$ pour une entité donnée :

- A est la matrice carrée de taille $N \times N$ qui donne la probabilité a_{ij} de passer d'un état à un autre indépendamment du temps, car c'est un système stationnaire ;
- B est la matrice de taille $N \times M$ qui fournit la probabilité d'émettre une observation dans un état q_i avec la particularité d'avoir la somme des observations pour un état donné égal 1.
- π_i est la distribution initiale de chaque état.

Globalement, lorsque l'on cherche à reconnaître une entité, par exemple un mot (une ligne ou un caractère) grâce à un système de HMMs, il faut calculer pour chaque modèle λ_i celui qui maximise la vraisemblance pour une suite d'observations données. L'ensemble des probabilités mentionnées sont représentées sur la figure 2.3.

L'apprentissage des paramètres se fait automatiquement grâce à un algorithme efficace *Baum-Welch*. Ce dernier est un cas particulier de l'approche *Expectation-*

	HMM	GMM
N	# d'états	
M	# d'observations	# de gaussiennes
A [1]	$A = \{a_{ij}\}$ où $a_{ij} = \mathbb{P}(q_t = s_j \mid q_{t-1} = s_i)$	
B [2]	$B = \{b_j(k)\}$ où $b_j(k) = \mathbb{P}(o_t = v_k \mid q_t = s_j)$	$b_j(O_t) = \sum_{j=1}^{M} C_{jm}\, \mathcal{N}(O_t, \mu_m, U_{jm})$
π	Distribution des états à l'état initial	

TABLE 2.1 – Définition des paramètres pour un modèle discret (HMM) et un modèle continu (GMM)

Maximisation permet de maximiser la vraisemblance pour chaque modèle λ_i. Il utilise également un autre algorithme connu qui est *Forward-Backward* qui permet de calculer les probabilités d'observations dans les états cachés. L'algorithme récursif *Forward* calcule la probabilité $\alpha_t(i)$ d'observer une séquence dans un état donné à l'instant t sachant qu'il existe N chemins pour arriver dans cet état ; tandis que l'algorithme *Backward* calcule la probabilité $\beta_{t+1}(j)$ d'observer une séquence dans un état supposé à l'instant t. Ce dernier principe parcourt le modèle de la fin vers les états initiaux. L'équation 2.1 est utilisée pour calculer la probabilité d'être dans un état s_i à l'instant et dans l'état s_j l'instant suivant avec la participation de la probabilité 2.2. L'équation 2.3 permet de calculer la probabilité d'être dans un état s_i à l'instant t. Les paramètres a_{ij} et $b_j(k)$ du modèle λ sont ré-estimés en combinant les résultats obtenus pour les équations 2.1 et 2.3, cela correspond à l'étape de Maximisation de *Baum-Welch*.

$$\xi_t(i,j) = \mathbb{P}(q_t = s_i, q_{t+1} = s_j \mid O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{\mathbb{P}(O \mid \lambda)} \tag{2.1}$$

$$\mathbb{P}(O \mid \lambda) = \sum \alpha_t(i) \beta_t(i) \tag{2.2}$$

$$\gamma_t(i) = \mathbb{P}(q_t = s_i \mid O, \lambda) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.3}$$

Une fois l'estimation de l'ensemble des paramètres opérée, l'algorithme de *Viterbi* est appliqué pour décoder une séquence donnée. Pour cela, il sélectionne le chemin optimal à travers les états afin de produire la séquence donnée.

Comme cela a été évoqué précédemment, les modèles HMMs peuvent être discrets, continus ou bien encore semi-continus. Cela se traduit concrètement par une différence sur la méthode utilisée pour émettre les probabilités d'observation d'une entité. Les modèles HMM continus utilisent des gaussiennes pour estimer

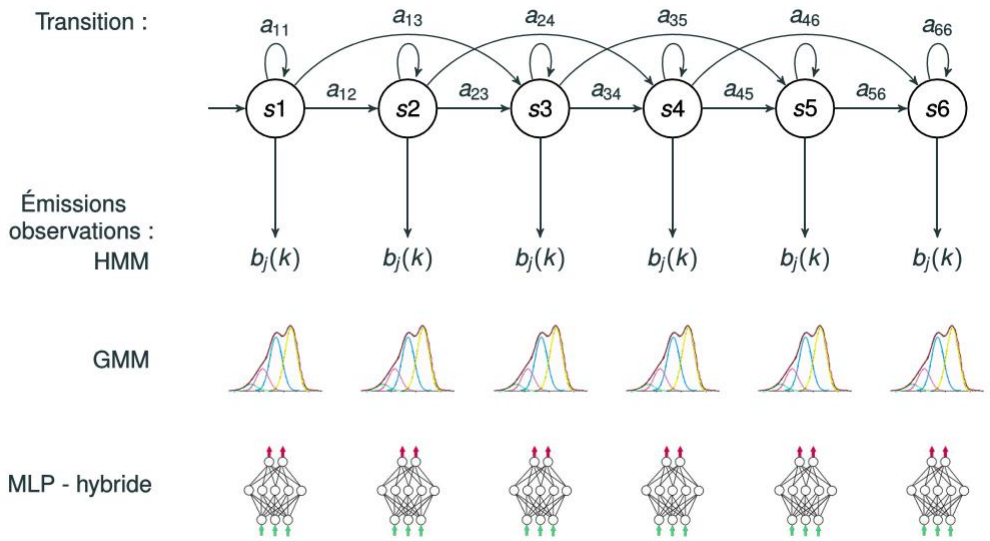


FIGURE 2.3 – Modèle de Markov caché discret, continu et hybride présentant un type "gauche-droite" ainsi que les probabilités à estimer.

les probabilités d'émissions des états de HMM comme le montre la figure 2.3. La table 2.1 permet de comparer un système discret et continu. La différence notable entre une chaîne continue et semi-continue est que la dernière partage les paramètres c'est-à-dire qu'un ensemble de gaussiennes est défini et chaque état partage cet ensemble. Cela permet d'éviter le sur-apprentissage. Il existe également des systèmes dits hybrides combinant des réseaux de neurones types *MultiLayer Perceptron* (MLP) et des modèles de Markov. Les réseaux de neurones sont utilisés pour calculer les probabilités d'émission des états. Le nombre de neurones dans la couche d'entrée correspond au nombre de caractéristiques utilisées et extraites sur les données. Ils sont généralement constitués d'une ou deux couches cachées avec une quantité de neurones supérieure à la couche d'entrée. Pour le reste, les modèles hybrides et continus sont similaires. Toutes ces méthodes sont présentées sur la figure 2.3.

Dans la reconnaissance d'écriture manuscrite, deux approches se retrouvent avec les différents modèles de HMM : une approche dite globale avec un système de modèles comportant autant de modèles que de mots ; ou une approche analytique avec un modèle par caractère, ces modèles peuvent ensuite être concaténés pour former des modèles de mots. Cependant, il est nécessaire de porter une attention particulière à la segmentation et le nombre d'états par modèles. En effet, si le

modèle avec un découpage en graphème a été choisi, cela permettra d'avoir une grande quantité et flexibilité de mots reconnus du lexique, mais cela implique un souci sur les ligatures entre les caractères (un caractère suivant sa position dans le mot aura des ligatures changeantes). Dans le cas d'un modèle de mots, le nombre de mots reconnus sera forcément limité, mais permettra une extraction plus précise de mots clés dans un document qu'avec une concaténation de modèles de lettres.

2.2.2 Approches neuronales profondes

Réseau de neurones récurrents (RNN) Depuis quelques années, les réseaux de neurones récurrents (RNN) révolutionnent le domaine. Ces méthodes discriminatives présentent un grand nombre d'avantages, comme une forte robustesse au bruit, et ils n'ont pas besoin de connaissances a priori. La récurrence de ce classifieur se matérialise par la liaison entre la sortie de chaque neurone n à l'instant $t-1$ à sa propre entrée à l'instant t comme le montre la figure 2.4. La mémoire du neurone (c'est-à-dire son activation) lui permet de prendre en considération le contexte ce qui a pour effet d'augmenter les performances des RNN. L'apprentissage des poids du réseau se fait par la méthode de la rétro-propagation du gradient (Werbos 1990) de manière itérative. Pour cela, l'erreur du gradient est calculée et propagée à travers le réseau pour l'ensemble des neurones. L'objectif est de converger vers un minimum global (idéalement). Cette méthode présente un problème majeur. En effet, à long terme, le gradient diminue rapidement au point de disparaître (Hochreiter et al. 2001), ce phénomène est appelé *Vanishing Gradient*. La conséquence sur le réseau est qu'il aura une tendance à ne prendre en compte que le contexte proche, ce qui est un problème en reconnaissance d'écriture où les images de mots et de lignes sont longues. Nous pouvons également relever un autre problème lié à l'apprentissage des poids qui sont en grande quantité dans ce type de réseau. Cela implique d'avoir une quantité de données suffisante afin d'éviter le risque de sur-apprentissage.

Définition de nouvelles cellules Pour résoudre le problème de "la disparition du gradient", (Hochreiter et Schmidhuber 1997a) ont proposé un bloc mémoire appelé *Long Sort-Term Memory* (LSTM). Ce bloc est présenté sur la figure 2.5. Il est composé d'une ou plusieurs cellules centrales C, leurs états sont contrôlés par trois types de portes (des portes d'entrée i_t, de sortie o_t et d'oubli f_t) qui sont elles-mêmes connectées au(x) cellule(s). Ces portes permettent de synchroniser les différentes entrées et sorties d'un état t. La porte d'entrée a pour fonction de contrôler si la nouvelle information doit être prise en compte dans le calcul de l'activation de la cellule ; la porte de sortie contrôle si l'activation de la cellule est propagée dans la sortie ; et la porte d'oubli contrôle la mémoire de la cellule et permet

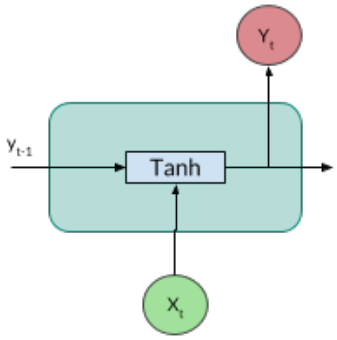


FIGURE 2.4 – Architecture interne d'un neurone classique utilisé dans un réseau récurrent.

de la réinitialiser si besoin est. Ce type de cellule est efficace pour les systèmes ayant besoin du contexte temporel à long terme. Cependant, l'apprentissage d'un réseau composé de cellules LSTM nécessite toujours d'avoir une grande quantité de données.

Un autre type de cellule *Gated Recurrent Unit* (GRU) a été proposé par (CHO et al. 2014a). Cette cellule est plus simple qu'une cellule LSTM, elle possède moins de paramètres à apprendre comme le montre la figure 2.6. Elle ne possède pas de mémoire, mais exploite directement les informations des états précédents. Elle est constituée de deux portes, z_t pour faire une mise à jour du contenu et r_t pour ré-initialiser et ainsi faire oublier le précédent calcul. L'utilisation de cette cellule permet d'accélérer la vitesse d'apprentissage dû à la différence de paramètres à calculer. Cependant, différentes études ont comparé l'utilisation de ces deux cellules sans parvenir à une véritable conclusion (CHUNG et al. 2014 ; IRIE et al. 2016), cela dépend de la tâche à réaliser et de la quantité de données disponibles.

Réseaux de neurones xDimensionnels Que ce soit avec les modèles de Markov cachés ou les réseaux de neurones récurrents, ils prennent en entrée un signal à 1 dimension, ne prenant en compte que le contexte passé. Une autre manière d'améliorer les performances des systèmes est de considérer l'information passée *forward* (comme c'était déjà le cas), mais également future appelée *backward* (SCHUSTER et PALIWAL 1997). Cela signifie que dans le cadre de la reconnaissance d'écriture manuscrite que la séquence d'entrée est lue de gauche à droite c'est-à-dire suivant le sens de l'écriture ainsi que de droite à gauche c'est-à-dire dans le sens inverse. Pour un caractère donné, une décision pourra être prise prenant en compte les caractères précédents et suivants. Ces réseaux sont appelés "bidirectionnel". D'un

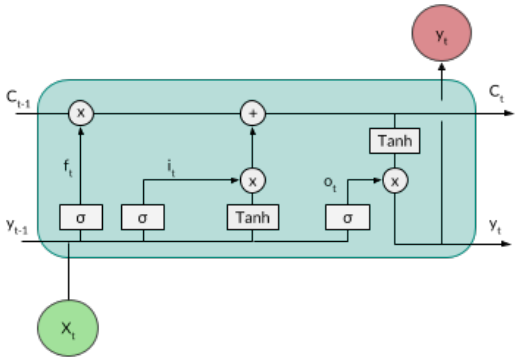


FIGURE 2.5 – Architecture d'une cellule LSTM.

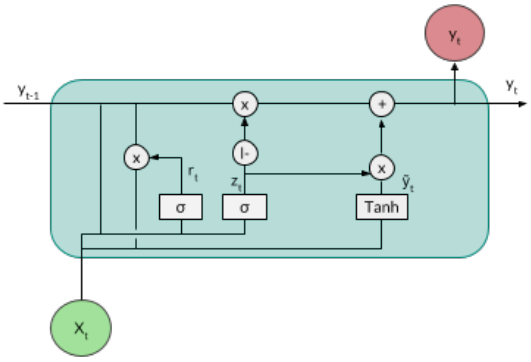


FIGURE 2.6 – Architecture interne d'une cellule GRU

point de vue architectural, cela se modélise avec deux couches cachées de neurones récurrents (l'une avant et l'autre arrière) indépendantes durant la phase d'apprentissage. Puis, les activations des deux couches sont combinées dans la couche de sortie. La figure 2.7 montre un exemple d'un réseau de neurones bidirectionnel utilisant des cellules LSTM (BLSTM). Le fonctionnement est exactement le même pour de simples neurones récurrents, le modèle est alors appelé BRNN. Les modèles BLSTM ont montré de meilleurs résultats en reconnaissance d'écritures par rapport aux HMM comme le prouve l'article (GRAVES et al. 2009) ainsi que (FISCHER et al. 2009b) qui ont testé cette méthode sur de la reconnaissance d'écritures manuscrites médiévales *offline*. La combinaison du passé et du futur semble rendre le système plus stable.

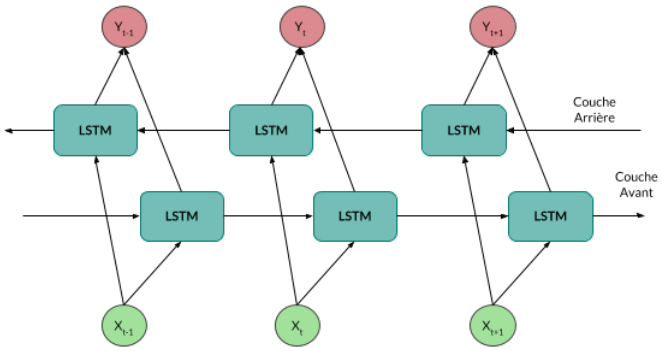


FIGURE 2.7 – Architecture d'un réseau bidirectionnel composé de cellules LSTM.

Une des dernières architectures, devenue une référence, est l'utilisation de réseaux multidimensionnels. Le signal d'entrée n'est plus seulement parcouru horizontalement, mais exploite toutes les dimensions disponibles de 2 à n. Une couche

cachée est alors dédiée à chaque sens de parcours sur le signal d'entrée. Pour la reconnaissance d'écriture manuscrite, l'image peut être utilisée directement en entrée du réseau sans passer par l'étape d'extraction des caractéristiques préliminaire. Ainsi, l'image est parcourue horizontalement et verticalement par le biais de quatre couches cachées. Cela permet pour un pixel donné d'avoir tout son contexte spatial indiquant sa position dans l'image. Depuis que ces architectures de type MDRNN et MDLSTM sont apparues (Graves et al. 2007 ; Graves et Schmidhuber 2009), elles dominent les compétitions en reconnaissance d'écriture manuscrite (Grosicki et El Abed 2009b ; Moysset et al. 2014). D'autres approches les combinent avec de multiples convolutions (Pham et al. 2014 ; Granell et al. 2018) favorisant plus largement l'extraction de caractéristiques.

Décodage par CTC Dans le cadre de la reconnaissance d'écriture, les réseaux que nous avons présentés jusque-là, bien qu'ils soient tous performants, ne permettent pas de faire l'alignement entre la séquence d'entrée et les étiquettes de façon automatique. Les réseaux de neurones récurrents sont utilisés pour faire de la classification temporelle de par leur structure (mémorisation de ce qui s'est passé à $t-1$). Mais, ils ne permettent pas encore d'atteindre l'étape d'étiquetage d'une séquence donnée sans passer par une étape de pré-segmentation des données en caractères et de post-traitement pour étiqueter. Pour résoudre cela, initialement dans un contexte de reconnaissance de la parole, (Graves et al. 2006) ont proposé un algorithme appelé *Connectionist Temporal Classification* (CTC) qui permet un étiquetage de la séquence sans avoir connaissance au préalable de l'alignement. Globalement, il autorise le système à ne pas prendre à chaque instant de décision sur un caractère potentiel en intégrant un joker dans les sorties possibles. Il s'applique sur la couche de sortie de type *Softmax* du réseau de neurones BLSTM comme à son origine, ou même MDLSTM.

La couche de sortie est constituée de $A+1$ étiquettes (neurones) où A est un alphabet (par exemple : lettres latines, espace, et ponctuations). L'étiquette supplémentaire est un joker appelé *blank*, qui peut être utilisé pour séparer des caractères qui se répètent, pour différer une décision, ou simplement pour ne pas répéter une étiquette. À chaque instant t, l'ensemble des étiquettes ont une probabilité conditionnelle jointe dans laquelle y_k^t est la probabilité pour une étiquette à un instant donné. Cela s'interprète comme la probabilité d'observer cette étiquette à un instant t pour une séquence d'entrée x de longueur T. Finalement, la probabilité d'avoir une séquence d'étiquettes π est calculée pour une séquence X avec la probabilité y d'observer une étiquette à un instant t :

$$\mathbb{P}(\pi|x) = \prod_{t=1}^{T} \mathbb{P}(\pi_t|x,t) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{2.4}$$

Une entrée donnée peut être décrite par plusieurs séquences différentes. Pour résoudre cela, une fonction surjective B est définie pour supprimer les répétitions de caractères non séparées par un *blank* puis pour supprimer les *blank*. Par exemple, la séquence $\{\_\_\_SSS\_o\_ppp\_hh\_i\_\_\_ee\_\_\_\_\}$ devient $\{Sophie\}$. Finalement, la probabilité de la séquence d'étiquettes finale l est calculée à partir de l'ensemble des chemins possibles :

$$\mathbb{P}(l|x) = \prod_{\pi \in B^{-1}(l)} \mathbb{P}(\pi|x) \tag{2.5}$$

Le CTC nécessite une modification de la phase d'apprentissage du réseau récurent en utilisant un algorithme *Forward-Backward* similaire à celui des HMM, mais en intégrant un treillis afin d'intégrer le label *blank* au sein de l'ensemble des étiquettes (au début, fin et entre chaque caractère). Il prend ainsi en considération l'ensemble des séquences possibles correspondant à la séquence d'entrée.

Pour reconnaitre une séquence de caractères à partir d'un BLSTM déjà appris, il faut pouvoir décoder la sortie du réseau. Il existe plusieurs méthodes de décodage possibles :

— le décodage par le meilleur chemin qui consiste simplement à sélectionner la sortie la plus active à chaque instant t, puis à toutes les concaténer en utilisant la fonction bijective B pour finalement obtenir la séquence la plus probable ;

— le décodage par recherche de préfixe calcule par récurrence la probabilité λ d'observer une étiquette à un instant t sachant le préfixe p. Cette méthode permet de prendre une décision sur la séquence d'étiquettes en prenant en compte plusieurs instants.

Ces deux méthodes admettent l'utilisation de dictionnaires et de modèles de langues pour assurer que les séquences d'étiquettes correspondent bien à des mots (et suite de mots). Les différentes méthodes de décodages, mais également de l'apprentissage adapté du *forward-backward* sont détaillées dans (Graves 2012b).

Cette sur-couche, combinée à un réseau type BLSTM, a montré des performances très supérieures aux systèmes hybrides comme le montre A. Graves dans ses articles que se soit en reconnaissance de la parole, ou de l'écriture (Graves et al. 2009 ; Graves 2012a). Depuis, cet algorithme CTC a été adopté dans le domaine de la reconnaissance d'écriture (Xie et al. 2016 ; Mioulet et al. 2015). D'autres études, comme celle de Bluche et al. 2015, ont comparé l'utilisation d'une topologie de type CTC sur des systèmes HMM hybrides et MLP. Ils ont ainsi mis en évidence que l'utilisation du *blank* permettait d'améliorer les performances dans le cas d'un système avec peu d'états et uniquement dans ce cas-là. Et, dans le cadre

d'un réseau récurrent, le *blank* permet d'éviter des problèmes de transcriptions au début de l'entraînement et autorise plus d'associations pour une même séquence. Ils concluent sur le fait que cette méthode est plus performante avec un réseau récurrent qu'un autre système. Le point négatif de ce type de système est que le temps d'apprentissage des réseaux récurrents est plus grand que pour un système de chaînes de Markov, mais cela est compensé au moment du décodage où il prend largement le dessus.

2.3 Modélisation du langage

Les résultats produits par les réseaux que nous avons présentés ne sont pas parfaits. Il est important de leur apporter de l'aide afin que les séquences produites aient du sens, que ce soit au niveau mot ou au niveau ligne. L'idée est de contraindre la reconnaissance d'écriture manuscrite. Le traitement automatique du langage (TAL) est alors utilisé pour renforcer les modèles. Il permet d'apporter des solutions pour exploiter les documents dans différents domaines comme la recherche d'information, la génération de texte automatique, la traduction automatique, mais également pour traiter des documents multimédias comme en reconnaissance de la parole et de l'écriture manuscrite. Le TAL peut aider à améliorer le décodage aux niveaux mots et caractères en modélisant un vocabulaire pour une langue donnée. Il existe plusieurs stratégies utilisées en reconnaissance d'écriture manuscrite comme les dictionnaires et les approches statistiques.

2.3.1 Les dictionnaires

Cette technique simple consiste à construire un dictionnaire de mots possibles dans la langue considérée. Cependant, la tâche de décodage du modèle est directement dépendante de la qualité du dictionnaire. Il est possible de le construire en se basant directement sur le vocabulaire des documents étudiés, d'utiliser un dictionnaire existant de la langue ou plus largement encore une ressource telle que Wikipédia. La limite avec cette solution est la proportion de mots hors-vocabulaire qui ne pourront pas être reconnus avec un dictionnaire petit ; et inversement, l'utilisation d'un large dictionnaire dispersant le modèle en proposant trop de mots similaires. Cependant, l'évolution de la langue et du vocabulaire fait qu'il y aura toujours des mots hors-vocabulaire et rend l'utilisation d'un dictionnaire difficile pour les tâches de reconnaissances. Le calcul de la couverture lexicale permet de mesurer l'efficacité d'un dictionnaire, en comptant le nombre de mots communs entre le dictionnaire et le corpus de test divisé par la taille du vocabulaire de

test. Cela donne une limite haute que le système peut atteindre en utilisant ce dictionnaire sur le corpus.

2.3.2 Les modèles de langage probabilistes

Un modèle de langage probabiliste donne une probabilité à une séquence S pour indiquer si elle existe ou non par rapport à un langage. En utilisant la règle de probabilités en chaîne, cette dernière se calcule comme étant le produit de probabilités conditionnelles en utilisant l'historique $h_i = x_1 \dots x_{i-1}$ de chaque élément x_i qui se note :

$$P(S) = P(x_1) \times P(x_2|x_1) \times \dots \times P(x_N|x_1 \dots x_{N-1}) = \prod_{i=1}^{N} P(x_i|x_1 \dots x_{i-1}) = \prod_{i=1}^{N} P(x_i|h_i).$$

L'historique de chaque élément doit être calculé sur un large corpus. Les modèles de langage peuvent être utilisés pour définir des séquences de mots ou de caractères.

Les n-grammes Le modèle le plus utilisé pour aider dans les différents domaines est un modèle de langage de type n-grammes. Il limite l'historique h_i d'un élément x_i aux $n-1$ éléments qui le précèdent. La séquence de n éléments est appelée un n-gramme où n désigne l'ordre du modèle. La probabilité d'une séquence s'écrit alors :

$$P(S) = \prod_{i=1}^{N} P(x_i|x_{i-n+1}^{i-1}).$$

Ces probabilités sont estimées en calculant la fréquence relative des n-grammes sur un corpus d'apprentissage soit

$$P(x_i|x_{i-n+1}^{i-1}) = \frac{c(x_{i-n+1}^{i-1} x_i)}{\sum_{x_j \in V} c(x_{i-n+1}^{i-1} x_j)}.$$

Pour calculer la probabilité d'un n-gramme, le nombre d'occurrences, désigné par la fonction $c(.)$, du n-gramme dans le corpus divisé par l'ensemble des n-grammes qui partagent le même historique.

Les n/m-grammes Initialement proposé par (Deligne et Bimbot 1997) pour la reconnaissance de la parole, le modèle multigrammes se base sur le fait qu'une phrase puisse se construire à partir de différentes séquences de mots de longueurs variables jusqu'à m. Cela signifie que le calcul de la probabilité d'une séquence est le produit

des probabilités de toutes les séquences possibles faisant varier la tête de 1 à m mots et l'historique de 1 à $m \times (n-1)$ mots. Récemment, les auteurs de (Swaileh et al. 2017b) ont montré que cette méthode était efficace en reconnaissance d'écriture manuscrite pour résoudre les problèmes de mots hors-vocabulaire.

Lissage des probabilités Même si le corpus d'apprentissage sélectionné est grand, il se peut que le calcul du n-gramme soit nul si le n-gramme n'est pas représenté. Pour éviter ce genre de cas, des techniques de lissage sont mises en place pour éviter des probabilités nulles, elles combinent une réduction et une redistribution des probabilités.

Réduction absolue opère une réduction identique sur l'ensemble des probabilités du corpus indépendamment de leur valeur.

Réduction relative opère une réduction de la probabilité en fonction de la fréquence du n-gramme.

Repli ou *Backoff* (Katz 1987) utilise les probabilités d'un modèle de langage d'ordre inférieur pour les n-grammes non observés au cours de l'apprentissage. En d'autres termes, si le n-gramme n'a aucune observation dans le modèle, la probabilité du (n-1)-gramme est utilisée. Ce sens hiérarchique permet de se diriger vers un modèle plus large et général, quand le modèle spécifique ne donne pas de résultats.

Interpolation réalise une combinaison des probabilités des différents modèles de langage d'ordre inférieur. Cette combinaison utilise un paramètre λ qui doit être optimisé pour réduire et redistribuer les probabilités efficacement.

Laplace considère que chaque n-gramme a été vu au moins une fois dans le corpus ce qui se traduit par $P(x_i|x_{i-n+1}^{i-1}) = \frac{c(x_{i-n+1}^{i-1}x_i)+1}{\sum_{x_j \in V} c(x_{i-n+1}^{i-1}x_j)+|V|}$.

Évaluation du degré d'incertitude Pour évaluer l'efficacité d'un modèle de langage par rapport au corpus de test T, il faut calculer la perplexité PPL. Elle correspond au nombre moyen d'hypothèses qu'a le modèle à chaque fois. La perplexité se calcule en utilisant l'entropie croisée du modèle sur le corpus T avec $P(T)$, le produit des probabilités de chaque mot du corpus de test :

$$PPL_{ML}(T) = 2^{H_{ML}(T)} \text{ où } H_{ML}(T) = -\frac{1}{|T|}\log_2(P(T)).$$

Il est important de noter qu'une fois de plus, la performance du modèle est directement liée au vocabulaire et au corpus d'apprentissage.

2.3.3 Approches neuronales

La représentation de modèle de langage par apprentissage a été proposée par (Bengio et al. 2003), plus connue aujourd'hui sous le terme de *word embedding* dont un schéma explicatif est donné en figure 2.8. Cette méthode consiste à représenter les mots du vocabulaire par un vecteur de caractéristiques en se basant sur un contexte défini via une fenêtre. Cela permet d'inclure la notion d'analyse sémantique dans la représentation de chaque mot. En effet, deux mots sémantiquement proches possèderont deux vecteurs de caractéristiques très similaires. Le réseau de neurones prend en entrée un vecteur de dimension $|V|$ (soit la taille du vocabulaire). Pour un mot donné, son historique est donné en entrée du réseau en indiquant 1 dans la ligne correspondant au mot. Finalement, la sortie du réseau fournit un vecteur de taille $|V|$ indiquant le prochain mot par la valeur 1 également. La représentation du modèle correspond à la partie interne du réseau qui nécessite un apprentissage. La matrice permettant de représenter les mots dans un espace vectoriel commun est de dimension $|V| \times N$ où le paramètre N est défini empiriquement, même s'il est souvent fixé à 300 caractéristiques comme pour les modèles *Word2Vec* (Mikolov et al. 2013) ou *FastText* (Bojanowski et al. 2016). Les modèles les plus récents comme *ELMo* (Peters et al. 2018) sont de plus en plus profonds et ajoutent des couches bidirectionnelles pour améliorer les performances. Les différents avantages d'une telle approche sont l'utilisation des poids partagés pour réduire les temps de calcul et la ré-utilisation de ces poids (*fine-tuning*), mais qui est également utilisée pour les approches classiques neuronales. En effet, des modèles dans différentes langues au niveau mot ou caractère ont déjà été entraînés sur de grands corpus, et leurs poids appris sont disponibles pour une utilisation directe.

En reconnaissance d'écriture manuscrite, (Zamora-Martinez et al. 2014) expérimente l'utilisation des réseaux de neurones pour modéliser le langage avec différents modèles de reconnaissance d'écriture manuscrite. Ils constatent que lorsqu'un large vocabulaire est utilisé, le réseau hybride HMM/ANN montre de meilleurs résultats que le réseau de neurones BLSTM. Cependant dans les deux cas grâce au modèle de langage neuronal, ils obtiennent d'excellents taux de reconnaissance sur les caractères et les mots. Une étude similaire a été menée sur de la reconnaissance de caractères chinois en comparant différentes structures de réseaux et avec différents niveaux de caractères (Wu et al. 2015 ; Wu et al. 2017).

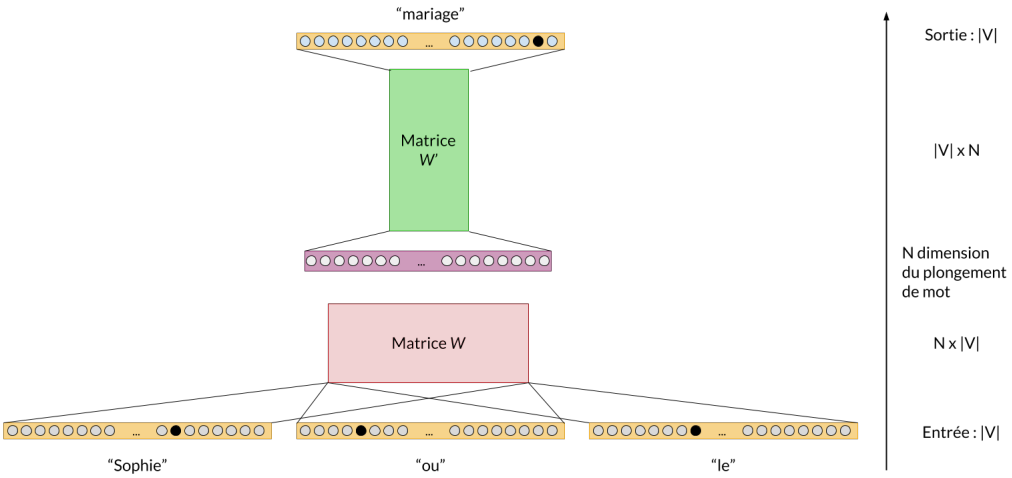


FIGURE 2.8 – Fonctionnement du *word embedding*. Les entrées sont le mot à l'instant t et les mots de son contexte. La sortie est la prédiction du mot à suivre.

2.4 Apprentissage par transfert de connaissances

Ces dernières années, l'utilisation des réseaux profonds d'apprentissage a explosé dans beaucoup de domaines. Cependant obtenir de bonnes performances avec ces systèmes a un prix. Ils requièrent beaucoup de données pour leur apprentissage ainsi que des ressources de calculs puissantes. Lorsqu'une nouvelle étude commence et nécessite la mise en place d'un nouveau réseau profond, il est impératif de prendre le temps de réfléchir à ces besoins. Le manque de données pour ce type d'apprentissage est un véritable frein. La constitution d'une nouvelle base pour répondre à cela est souvent très couteuse en temps pour collecter et annoter, ce qui est parfois impossible à réaliser. Un autre point est le cadre d'apprentissage de ces réseaux. Ils fonctionnent sur des données d'apprentissage et de test partageant des caractéristiques et provenant d'un même ensemble de données qui a été partitionné. Si les données de tests proviennent d'un autre ensemble, il faut réaliser un nouvel apprentissage afin d'obtenir un système dédié. La solution émergente couvrant ces difficultés est le *transfer learning* autrement appelé en français apprentissage par transfert de connaissances.

2.4.1 Définition de l'apprentissage par transfert de connaissances

L'apprentissage par transfert de connaissances consiste à entraîner un modèle pour une tâche pour ensuite l'appliquer pour une autre tâche liée. Le but est d'optimiser le modèle pour tendre vers une généralisation de ce dernier en utilisant des données et informations provenant d'un autre contexte. En répondant aux questions suivantes : quelle connaissance transférer ?, comment la transférer ? et quand la transférer ?, cela permet de différencier plusieurs formes de l'apprentissage par transfert de connaissances selon la disponibilité des données d'apprentissage et de test, ainsi que le domaine d'application.

Apprentissage par transfert inductif s'opère pour une tâche source différente de la tâche cible, mais liée, le domaine d'application qu'il soit différent ou identique dans cette configuration importe peu. Dans le cadre où il y a beaucoup de données annotées sources, nous parlons d'une configuration d'apprentissage multi-tâches dans laquelle les tâches sources et cibles sont apprises simultanément. Par opposition, une configuration d'apprentissage autodidacte doit réaliser l'apprentissage sans avoir de données sources.

Apprentissage par transfert transductif s'opère pour une tâche source et cible qui est identique, mais de domaines différents. Autrement dit, lorsqu'il n'existe pas de données cibles déjà annotées pour réaliser l'apprentissage mais que d'autres données sources de domaines différents sont disponibles. Suivant les applications, il faudra faire attention à l'espace des caractéristiques extraites entre les différents domaines sources et cibles.

Apprentissage par transfert non supervisé s'opère à travers des tâches et des domaines différents, mais liés. Il est principalement utilisé pour résoudre des tâches d'apprentissage non supervisé dans un domaine cible particulier.

2.4.2 Principales approches

Il existe différentes méthodes pour pouvoir réaliser un apprentissage par transfert de connaissances. Nous listons ici les plus courantes :

Transfert de modèle consiste pour une tâche donnée cible à créer un modèle pour une tâche connexe ayant des données annotées et disponibles. Une fois que le modèle a été entrainé, il est possible de l'utiliser partiellement ou totalement comme point de départ pour la tâche cible d'origine.

Transfert d'instance consiste à ré-utiliser des données provenant de l'apprentissage dans le domaine source pour l'apprentissage du domaine cible avec l'aide d'une re-pondération des poids du réseau.

Transfert de connaissances relationnelles consiste à rechercher les relations au sein des données du domaine source pour les appliquer dans celles du domaine cible.

Transfert de représentation des caractéristiques consiste à apprendre une "bonne" représentation automatiquement des caractéristiques sur les données du domaine source, et qui devrait obtenir des résultats similaires sur les données du domaine cible. Pour ce faire, il est d'usage de travailler avec des modèles déjà pré-entrainés en sélectionnant certaines couches particulières du réseau. Dans ce type d'approche, les poids optimisés du réseau pré-entrainé peuvent être fixés pour ne pas être altéré lors de l'apprentissage éventuel du nouveau réseau ou bien ajustés avec un très faible taux d'apprentissage pour ne pas désapprendre les connaissances acquises précédemment. Une autre solution de plus en plus mise en œuvre est la réalisation de système auto-encodeurs à convolution par exemple, qui lors de son apprentissage va avoir pour objectif dans un premier temps de reproduire la séquence d'entrée. Cette approche permet de fournir des caractéristiques généralistes et qui ne change pas avec le domaine. Globalement, le transfert de caractéristiques est très utilisé en traitement du langage avec les *Word Embedding* ou modèles de plongement de mots ainsi qu'en traitement d'images.

Transfert de paramètres consiste à apprendre des poids partagés ou des hyper-paramètres pour les tâches sources et cibles. Il est également possible d'utiliser des modèles pré-entrainés cette fois-ci. Mais en les utilisant pour initialiser le nouveau réseau dédié à la tâche cible, et de continuer l'apprentissage de l'ensemble du nouveau modèle.

2.4.3 Domaines d'applications et adaptations

L'apprentissage par transfert de connaissances est utilisé depuis quelques années et a montré de bonnes performances dans différents domaines. Que ce soit dans les domaines liés au traitement du langage ou en *Vision Object*, le nombre de modèles neuronaux entrainés, présentant de bonnes performances et disponibles, a explosé.

Dans le domaine du traitement de l'image, une pratique courante est d'utiliser les réseaux pré-entraînés en classification d'images sur les images d'ImageNet qui ont montré de bons résultats comme Alexnet, VGG ou encore GoogleNet sur d'autres tâches (Oquab et al. 2014 ; Donahue et al. 2014 ; Sun et al. 2015 ; Hu et al. 2015a). Souvent les données annotées ne correspondent pas aux données cibles à traiter. Par exemple, les images annotés d'objets isolés sur un fond uniforme sont disponibles mais les études portent sur la composition d'images où il y a plusieurs objets et du bruit. La modèle va être capable de contourner ces différences.

Il en est de même en traitement du langage où l'utilisation des modèles de plongement est devenue récurrente ces dernières années. Certaines études comme (Conneau et al. 2017) expérimentent directement des modèles universels pour les appliquer à douze tâches différentes, dont la détection automatique de paraphrase ou la recherche d'image. Il est plus difficile en traitement du langage d'opérer de l'adaptation suivant la tâche cible. Par exemple, si l'étude porte sur des tweets, il peut être difficile d'utiliser des modèles pré-entraînés sur des informations journalistiques ou wikipédia. Dans d'autres études par contre comme l'analyse de commentaires cela est plus facile car le vocabulaire d'opinion reste sensiblement le même en dehors des différents termes se référant à l'objet de l'étude changent. Pour finir sur le traitement du langage, comme nous l'avons déjà évoqué le *word embedding* est très large étudié et utilisé en apprentissage par transfert de connaissances cependant nous pouvons mentionner une des principales difficultés qui est le domaine d'application. Si ce dernier est très spécifique, il se peut qu'un large corpus, généraliste ne contient pas assez de termes liés à ce domaine.

Dans (Lladós et al. 2012), les auteurs étudient le problème de détection automatique de mots, appelé *word spotting*, pour des documents historiques. Ils s'orientent dans cette démarche, car elle semble moins coûteuse pour pallier une absence de données pour l'apprentissage. De plus, toujours selon les auteurs, l'utilisation d'un ensemble de données non dédié pour faire de la reconnaissance d'écriture est déconseillée car il peut y avoir plusieurs problèmes causés par des différences significatives en termes de période et de zone géographique, qui affectent souvent le style de l'écriture.

La recherche d'un ensemble de données d'apprentissage répondant aux caractéristiques spécifiques souhaitées pour la détection automatique de mots clés ou la reconnaissance d'écriture manuscrite est une tâche complexe. Il existe d'autres études qui tentent d'utiliser sur des documents historiques des systèmes appris sur des documents anciens. Dans le cadre de la détection automatique de mots (*Keyword Spotting*), les auteurs (Frinken et al. 2010) mélangent différentes ressources principalement contemporaines pour transcrire une ressource historique qui a peu de données, en faisant attention au vocabulaire utilisé dans les ressources d'apprentissage et de test, en d'autres termes, ils réalisent un apprentissage par transfert de connaissances.

2.5 Conclusion

Dans ce chapitre, nous avons montré qu'en reconnaissance d'écriture manuscrite, il se dessine une tendance à l'utilisation des réseaux de neurones à chaque étape,

que ce soit pour l'extraction de caractéristiques, pour la réalisation des modèles eux-mêmes ou encore pour les modèles de langages. Ces réseaux permettent d'intégrer et de prendre en compte plus de contexte qu'auparavant tout en le gardant en mémoire. Malheureusement, ils restent gourmands en ressources d'apprentissage.

Nous avons présenté les différentes étapes qui se succèdent. Une fois les documents numérisés, un certain nombre de pré-traitements peuvent être réalisés pour favoriser l'efficacité des systèmes de reconnaissance d'écriture manuscrite comme le nettoyage de l'image, les corrections des lignes et de l'écriture. D'autres étapes comme la segmentation en paragraphes, en lignes ou en mots sont également possibles, mais n'ont pas été évoquées. En effet, pour notre étude, nous avons choisi de minimiser le nombre de pré-traitements et de laisser la segmentation des images à une autre équipe. L'extraction de caractéristiques reste une étape déterminante, car elle conditionne la représentation des informations de l'image donnée au modèle de reconnaissance d'écriture manuscrite qui est l'élément pivot dans un système *end-to-end*. Pour améliorer les performances de ces systèmes et les guider, les modèles peuvent s'appuyer sur des outils du traitement du langage comme les dictionnaires ou les modèles de langage. Ce domaine connaît également une forte attraction vers les modèles neuronaux où ils dominent l'ensemble des tâches.

Dans nos expérimentations présentées dans les chapitres suivants, nous avons choisi de nous concentrer sur la réalisation d'un système de reconnaissance d'écriture manuscrite capable d'opérer un apprentissage par transfert de connaissances. Comme nous l'avons présenté, cela est déjà utilisé dans différents domaines, que ce soit en traitement du langage ou en *Vision Object*, afin de maximiser l'ensemble des connaissances acquises. Dans le chapitre 4, nous avons réalisé un système constitué d'éléments de l'état de l'art comme l'utilisation d'un RNC et d'un modèle BLSTM-CTC.

3

Ressources existantes et collectées

Sommaire

Introduction

Un des facteurs déterminants dans la reconnaissance d'écriture, qui va influer sur les résultats du système, est les ressources utilisées parmi celles qui sont disponibles pour l'apprentissage, mais également les ressources que l'on souhaite reconnaître. Comme nous l'avons indiqué en introduction, les documents que nous étudions de la Comédie-Italienne sont vierges de toutes analyses. Il faut donc commencer cette étude par la constitution d'une base autour des registres de la Comédie-Italienne, puis définir les ressources images et linguistiques répondant à certains critères pour réaliser l'apprentissage des systèmes.

Pour constituer une nouvelle ressource à partir de documents bruts, il existe trois solutions :

1. Demander à un expert d'annoter manuellement, mais cela est très coûteux en temps. Par exemple sur les registres de la Comédie-Italienne, il a été estimé qu'il faudrait 3 ans et demi en travaillant à temps plein, à un expert du domaine pour annoter l'ensemble des pages des registres. Cette solution n'est donc pas envisageable (voir la partie 3.1.2.3 pour plus de détails).
2. Utiliser des outils existants pour chaque étape : la classification des pages, la segmentation des zones, la reconnaissance d'écriture. Nous avons réalisé des expérimentations avec DMOS (Couasnon 2001) pour la détection et segmentation des zones dans les pages, mais la diversité des pages selon les registres considérés a été un frein à l'utilisation de cet outil.
3. Mettre en place un site d'annotation participatif (également appelé site de *crowdsourcing*).

C'est cette dernière solution qui a été privilégiée et réalisée par l'équipe DuKE du laboratoire LS2N. Nous présentons ici une partie de leur travail qui permet de mieux comprendre la source de la base d'images créée, ainsi que les traitements que nous avons appliqués sur leurs informations collectées. Ce travail collaboratif a été publié dans la conférence LREC en 2018 (Granet et al. 2018a).

Site participatif en quelques mots. C'est une plateforme en ligne [1] qui fait appel à des volontaires pour annoter des documents historiques selon des besoins définis au préalable. Dans notre cas, les besoins ciblés sont les suivants :

— définir le type de la page ;

1. http://recital.univ-nantes.fr/

— détecter les différentes zones et les identifier selon le type d'information qu'elles contiennent ;
— transcrire ces zones ;
— valider ou corriger les transcriptions fournies par les précédents utilisateurs.

Dans un souci de performance et de résultat, nous avons mis en place un système semi-automatique en parallèle de la plateforme afin d'accélérer le processus d'annotation et de validation. Ce système permet la segmentation automatique des zones en lignes ainsi que la validation des transcriptions associées à ces zones. L'approche globale est présentée dans la figure 3.1.

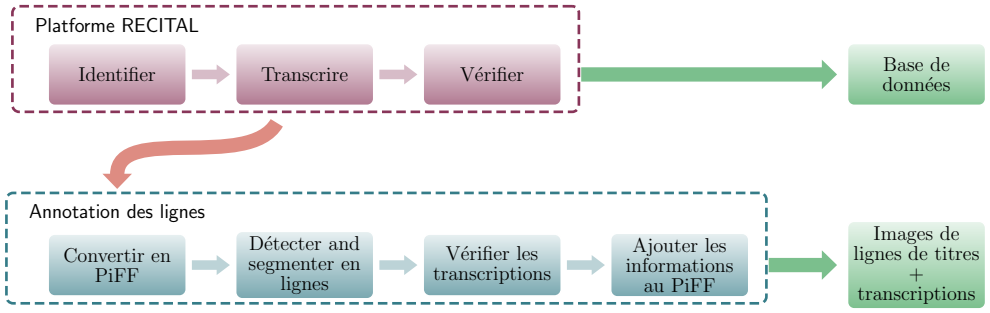


FIGURE 3.1 – Flux des données appliqué aux registres de la Comédie-Italienne.

3.1 RECITAL : Site de production participative pour les registres de la Comédie-Italienne

3.1.1 Travaux connexes autour des sites d'annotation participatifs

Dans le domaine de la reconnaissance d'écriture, certaines techniques et méthodes de l'état de l'art sont très connues comme les logiciels de Reconnaissance Optique de Caractères (ROC), communément appelés *OCR* pour *Optical Caracters Recognition*. Un de ses logiciels largement utilisés est Abby®FineReader. L'équipe DuKE l'a expérimenté sur une page de la Comédie-Italienne, mais les résultats obtenus n'étaient pas satisfaisants ni exploitables. Cela peut être dû aux spécificités énoncées dans la partie 1.3.

Réussir à extraire des informations de ces documents est une première étape. Il faut ensuite les sauvegarder pour pouvoir les exploiter. Pour cela, différents types de formats XML existent. Le format le plus connu et utilisé par les chercheurs en Sciences Humaine et Sociale est le format *Text Encoding Initiative* appelé TEI. Il

permet de structurer des documents textuels comme la transcription de pièces de théâtre. Il possède un grand nombre de balises ce qui complexifie son utilisation. Malgré tout, il n'est pas facile d'inclure la localisation de l'information dans la page ainsi que son type. Il existe un autre format *Page Analysis and Ground-truth Elements* (Pletschacher et Antonacopoulos 2010) appelé PAGE XML qui lui autorise la liaison entre les informations, les caractéristiques de l'image, la disposition ainsi que le contenu. Cependant, la diversité et la complexité des données contenues dans les pages journalières de la Comédie-Italienne exigent que les types et balises du format soient beaucoup plus précis et incluent les différents niveaux d'annotations que l'on peut avoir. C'est pour l'ensemble de ces raisons que nous nous sommes tournés vers un nouveau format *Pivot File Format* (Mouchere et al. 2017) appelé PiFF.

3.1.2 Présentation de la plateforme

Comme évoqué précédemment l'utilisation d'un OCR uniquement sur les registres de la Comédie-Italienne est impossible à cause de leur complexité. Tandis que les opérations de marquage et transcription sont des tâches dites *Human Intelligence Tasks* (Chittilappilly et al. 2016) qui sont adaptées pour le travail collaboratif.

La plateforme RECITAL [2] est une branche du projet ScribeAPI [3] existant, permettant de mettre en œuvre des tâches de labellisation et de transcription sur des documents résistant aux OCRs.

Les plateformes d'annotation participatives sont devenues un outil classique et montrent de bons résultats pour transcrire les documents dans les projets de sciences du numérique. RECITAL, quant à lui, doit intégrer des pré-traitements supplémentaires ce qui augmente son niveau de complexité. En effet, il est nécessaire de classer une typologie contenant plus de 100 catégories différentes liées principalement aux dépenses et revenus. Il faut, en plus, identifier les zones de titres, les dates, ainsi que beaucoup d'autres informations comme les noms des actrices et des acteurs...

3.1.2.1 Fonctionnement

Le processus séquentiel mis en place se compose de 3 activités différentes comme le montre la figure 3.2 :

2. http://recital.univ-nantes.fr
3. https://github.com/zooniverse/scribeAPI

1. **Marquage** : lors de cette étape, l'utilisateur doit classer la page montrée selon son type. 8 types différents sont répertoriés dont 3 (couverture, page blanche et inclassable) mettent fin au processus et retirent la page du circuit global. Puis, suivant le type de la page qui a été sélectionné, une succession de 5 écrans est proposée. Chaque écran comporte au plus 10 marques correspondant à des informations à identifier dans la page. Les utilisateurs peuvent annoter autant d'information qu'ils le souhaitent. À la fin de la séquence d'écrans, il leur sera demandé s'il reste des choses à annoter sur la page ou non. Il leur est également possible d'arrêter le processus d'annotation à tout moment.
2. **Transcription** : lors de cette étape, l'utilisateur doit transcrire les zones de textes identifiées à l'étape précédente. Il leur est possible d'indiquer si une marque est erronée, c'est-à-dire mal positionnée ou bien illisible.
3. **Vérification** : lorsque deux transcriptions différentes sont données pour une même marque, elles sont soumises à un vote des autres utilisateurs afin de trouver un consensus.

Le processus mis en œuvre dans RECITAL offre un très grand nombre de micro-tâches. En regroupant les 8 types de documents, nous comptabilisons 133 éléments différents qui peuvent être classifiés comme le montre la figure 3.2. De plus, l'utilisateur peut transcrire directement la marque qui vient juste d'être réalisée.

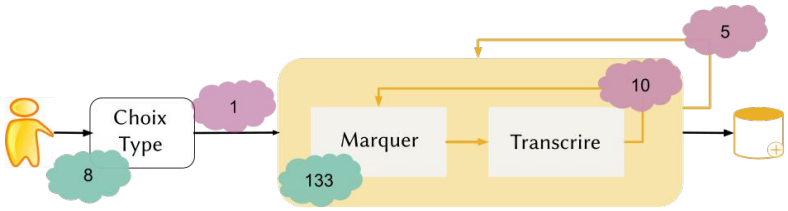


FIGURE 3.2 – Vue simplifiée de l'ensemble du processus sur RECITAL pour une page. L'étape de vérification n'est pas représentée. Les figures bleues représentent le nombre de tâches internes, et les figures violettes représentent le nombre de catégories pour chaque étape.

3.1.2.2 Configuration

Assignation des tâches L'utilisateur peut choisir la tâche qu'il préfère réaliser parmi le marquage, la transcription et la vérification. Une fois ce choix effectué, une tâche est sélectionnée aléatoirement et assignée à l'utilisateur parmi celles disponibles. Il est également possible pour l'utilisateur de sélectionner un registre particulier de la Comédie-Italienne sur lequel il souhaite travailler. La quantité de

pages suivant les années varie de 192 à 590. Pour finir, un utilisateur (s'il s'est authentifié au préalable) ne peut pas faire la vérification des transcriptions qu'il a lui-même réalisées.

Gestion des cas particuliers À chaque étape, l'utilisateur peut gérer les erreurs et les cas particuliers.

— au moment de l'identification du type de page, il peut indiquer la page comme inclassable ;
— au moment de la transcription, il peut indiquer si une marque a été mal placée par un autre utilisateur.

Une fois que deux utilisateurs différents indiquent la même erreur, l'élément en question (une page ou une marque) est retiré de la base.

Vers un consensus Chaque marquage est soumis au moins à deux utilisateurs différents pour être transcrit. Pour mesurer la correspondance entre plusieurs transcriptions proposées, des algorithmes de perte sont utilisés pour atteindre plus facilement un consensus, comme il est suggéré par (Matsunaga et al. 2016). De plus, ces algorithmes sont insensibles à la casse, ignorent les espaces et la ponctuation pour faciliter la mesure. Si les deux transcriptions sont identiques, la transcription candidate est acceptée, et la marque est retirée du processus. Dans le cas contraire, le marquage est soumis à nouveau à un vote. Au cours de la vérification, l'utilisateur peut soit choisir une transcription déjà existante soit en proposer une nouvelle. Dans le cas où plusieurs transcriptions ont été proposées, le consensus est atteint à la majorité votante. Le seuil de cette majorité est fixé à 75% des votants qui vont de 3 à 10 utilisateurs. Une fois que la limite d'utilisateurs est atteinte sans avoir obtenu la majorité, le vote est clos et l'annotation est classée comme dissensus.

3.1.2.3 Chiffres clés

La plateforme RECITAL a été ouverte au public en Février 2017. Les chiffres suivants sont extraits aux dates du 19 Juin 2017 et 23 Janvier 2018, date à laquelle une nouvelle analyse de quantitative a été réalisée.

Heures de travail Il y a en moyenne 30 informations à transcrire par page. En se basant sur des données existantes obtenues lors des premières opérations sur RECITAL, il a été possible de réaliser une estimation du temps passé par un expert du domaine de la Comédie-Italienne pour chaque information :

— 23,5 secondes pour le marquage ;
— 13 secondes pour les transcriptions ;
— 26 minutes par page.

En sachant qu'il est possible de faire 1 540 heures travaillées par an, il faudrait près de 3,5 ans pour qu'un expert à plein temps complète la transcription des 25 250 pages de cette collection.

Utilisateurs et réponses RECITAL héberge 25 250 pages sur les 27 544 pages existantes. À la date du 23 janvier 2018, 68 540 tâches (voir Figure 3.3) ont été effectuées par 314 travailleurs (au 19 juin, 625 utilisateurs étaient actifs sur la plateforme). Bien que ce nombre soit un indicateur de l'activité sur la plateforme et révèle l'engagement des utilisateurs, il ne constitue pas un bon aperçu des progrès réels dans l'étiquetage et l'annotation du corpus. Nous avons donc calculé (Figure 3.4) le nombre de marques, de transcriptions et de votes en attente (ou clos) par page pour chaque page de chaque registre (un registre par an). Grâce au tableau 3.1, nous pouvons constater que le nombre de contributeurs est en constante augmentation. La plateforme a fait régulièrement l'objet de présentation à différentes occasions, comme des conférences scientifiques ainsi que littéraires, et des cours sur le théâtre italien. Des journées d'annotations ont été régulièrement organisées depuis fin 2017, ce qui a permis d'accroître considérablement le nombre de marques sur la fin 2017 ainsi que le nombre de transcriptions validées au cours de l'année 2018. En ce qui concerne le profil des annotateurs, à notre connaissance, ce sont majoritairement des contributeurs liés au projet ou aux personnes directement impliquées dans ce dernier. Parmi les plus fréquents, nous retrouvons : l'ensemble des acteurs du projet ANR CIRESFI, mais également du Centre d'Études des Théâtres de la Foire et de la Comédie-Italienne, des étudiants de Master en lettres de l'Université de Nantes et de Lyon, ainsi que des scientifiques et chercheurs curieux. Il reste difficile d'établir l'ensemble des profils, car la création d'un compte n'est pas nécessaire pour contribuer.

Table 3.1 – Avancement du travail d'annotation sur RECITAL

	6 Sept. 2017	23 Janv. 2018	10 Avr. 2018	9 Déc. 2018
Contributeur	183	314	550	768
Marques réalisées	18 353	68 540	95 000	> 97 000
Transcriptions validées	214	536	3 800	4 601

Annotations et consensus Parmi les 46 504 marques créées, 43,6% ont été transcrites par au moins un utilisateur. Parmi celles qui ont au moins une transcription,

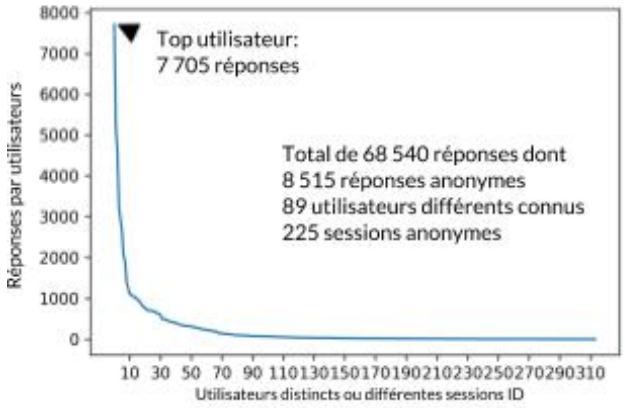


FIGURE 3.3 – Distribution des réponses représentant toutes les activités par travailleur à la date du 23 janvier 2018. Seuls les utilisateurs ayant accompli au moins une tâche complète sont pris en compte.

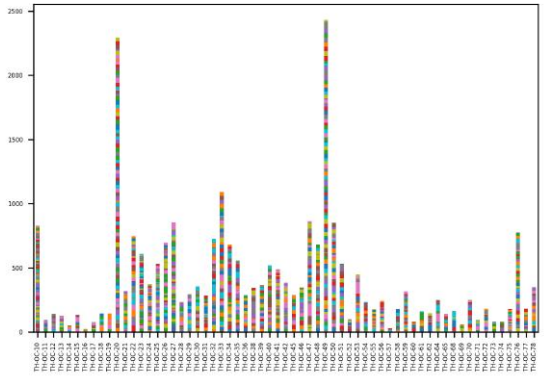


FIGURE 3.4 – Vue d'ensemble de l'avancement de RECITAL à la date du 23 Janvier 2018 pour chaque registre, ordonné par année. Chaque barre de couleur correspond à une unique page et la taille dépend du nombre de marques, de transcriptions et de consensus déjà réalisés dans cette page.

76,3 % ont été transcrites par un seul utilisateur, 11 % ont été complétées (dans les cas où 2 utilisateurs distincts ont produit la même transcription successivement), 10,1 % sont en attente de vote (avec au moins 2 transcriptions différentes), et 2,6 % (536 cas) ont abouti à un consensus (nous n'avons que 7 cas de dissensus, par exemple des échecs de consensus, dans notre ensemble de données).

3.1.2.4 Limites et perspectives

Au cours de la mise en œuvre de RECITAL, des limites ont pu être observées de même que des perspectives proposées dans la méthodologie employée.

Ordonnancement des registres La figure 3.4 présente le problème que pose le choix du registre au début du processus. Les registres du début du siècle en italien sont très peu annotés, ceci est dû à leur niveau de difficulté. C'est pour cette raison qu'une condition a été ajoutée sur un ordre arbitraire dans le processus pour ne pas effrayer les utilisateurs. Cet ordre est défini comme suit : les registres comptables après 1747 sont beaucoup plus faciles à marquer et à transcrire, il convient donc de les privilégier. Cela a permis de réduire le temps d'annotations des documents et de favoriser la fidélité des utilisateurs.

Libre arbitre Suivant les utilisateurs les transcriptions produites peuvent grandement varier. Cela va dépendre de leur expertise, mais également de leur jugement sur le résultat attendu. Par exemple, une date comme "*Du Mardy 12 Juillet 1768*" peut être transcrite comme un fac-similé, c'est-à-dire à l'identique, ou peut être interprétée et transcrite sous une forme canonique comme "*mardi 12/07/1768*". La normalisation est intéressante dans le cadre de post-traitements des données pour faciliter la recherche dans une base de données, mais elle est indésirable pour de la reconnaissance d'écriture. Nous pouvons lister un certain nombre de points pouvant poser des difficultés pour trouver des consensus comme les abréviations, la qualité approximative de l'écriture manuscrite, les multiples rédacteurs et les documents multilingues. Cela montre l'importance, le besoin ainsi que la difficulté de mettre en place des instructions, et de contrôler les entrées dans un tel processus participatif.

Convergence vers un consensus Les algorithmes de perte mesurant la similarité permettent de réduire la difficulté pour atteindre un consensus entre les transcriptions candidates. Cela corrobore ce qui a été montré dans l'article (Little et al. 2010) qu'un processus itératif permet d'atteindre plus facilement un consensus.

Implication des utilisateurs Dans ce genre de travaux, l'engagement des utilisateurs est une tâche difficile, mais nécessaire. Pour concrétiser cela, il est nécessaire de mettre en place des récompenses, des défis ainsi qu'un forum de discussion. Fournir des retours sur le travail que les utilisateurs ont effectué est une technique simple de motivation, mais efficace. (Clematide et al. 2016).

Malgré le travail produit par les utilisateurs, cela prend du temps de pouvoir trouver un consensus pour l'ensemble des transcriptions. Afin d'accélérer ce processus, nous avons combiné les premières transcriptions candidates obtenues sur RECITAL avec un système supervisé de segmentation et de validation.

3.2 Registres de la Comédie-Italienne

La création de ressources de test sur les données de la Comédie-Italienne est une étape importante pour évaluer les systèmes et vérifier qu'ils répondent au cahier des charges que nous nous sommes imposés. Deux stratégies différentes ont été mises en place pour les créer, une au début du projet, alors que la plateforme participative démarrait juste (sur un registre), puis une autre plus tard avec les premiers résultats (sur tous les registres).

3.2.1 Transcription des images

Les premières données collectées (même si elles n'ont pas encore été validées) sont utilisées pour simplifier cette tâche d'annotation. Nous nous sommes concentrés sur le champ de titre des registres de la Comédie-Italienne. Nous présentons les différents outils utilisés lors de la création de ces deux nouvelles bases.

Convertion au format PiFF Tout d'abord, nous avons sélectionné et converti toutes les données liées à la zone de titre dans le format PiFF [4]. Ce format est une solution pour pouvoir favoriser les échanges d'information entre les différents systèmes. Nous obtenons un fichier PiFF par page de registre. À chaque étape, le document est enrichi par les nouvelles informations collectées. Dans l'extrait de fichier fourni 3.1, la première partie correspond aux meta-données avec l'identifiant de la page dans les registres ainsi que l'URL sur Gallica. Cette partie reste constante tout au long des traitements. Dans notre cas, ce format nous permet d'enregistrer des polygones (ce qui correspond aux zones définies par les utilisateurs) donnés

4. https://gitlab.univ-nantes.fr/mouchere-h/PiFFgroup

dans la seconde partie, appelée “location”, à toutes les transcriptions candidates qui ont pu être faites, identifiés dans la dernière partie, appelée “data”.

Détection et segmentation en ligne Les coordonnées fournies par les utilisateurs sont utilisées pour extraire la zone de titre. Ensuite, nous y avons appliqué l'algorithme *Seams Carving*, appelé également TLA proposé par (Arvanitopoulos et Süsstrunk 2014) pour segmenter en ligne ce bloc de texte. Dans un premier temps, le bloc est découpé en petite fenêtre pour y calculer la ligne médiane par le biais d'une projection de pixel (voir Figure 3.6b). Dans un second temps, les lignes séparatrices de texte sont calculées selon un procédé similaire qui intègre une contrainte régionale grâce à un filtre gaussien (voir Figure 3.6c). Cette méthode a l'avantage de pouvoir être réalisée sur les images en niveaux de gris ; mais également, de favoriser le meilleur découpage sur les zones de chevauchement entre les hampes et jambages de lignes voisines. Nous l'utilisons lors de la création des deux bases sur la Comédie-Italienne.

Finalement, sans chercher à trier entre les lignes correctement segmentées et les erreurs, nous avons ajouté ces nouvelles coordonnées dans le fichier PiFF correspondant dans la section “polygon” avec un nouvel identifiant dérivant du bloc d'origine et un nouveau type (voir ligne 22 à 31 de 3.1). Chaque nouveau polygone est également associé aux transcriptions candidates faites sur la zone de titre de départ.

Vérification des transcriptions Notre but est de créer une vérité terrain pour des systèmes de reconnaissance d'écriture centrés sur les lignes de titre. C'est pour cela que nous avons construit une interface graphique simple (voir figure 3.5) pour valider les transcriptions sur les lignes de texte segmentées à l'étape précédente. Elle permet de vérifier si :

— le polygone présenté est une ligne de texte, et non une ligne vide, un filet ou une ligne mal segmentée ;
— la transcription associée correspond parfaitement à la ligne proposée.

Dans le cas où il existe plusieurs transcriptions candidates pour une zone de titre, elles sont toutes proposées pour sélectionner la meilleure ou corriger la plus proche de la vérité. Cela permet également de faire le tri entre les fac-similés et les transcriptions normalisées suivant le besoin. Une fois qu'une transcription est associée à une ligne, ces nouvelles informations viennent enrichir le fichier PiFF dans la partie “data”.

Extrait 3.1 – Exemple de fichier PiFF final pour la page TH-OC-50_0155.

```
{
 "meta": {
  "piff_version": "version 0",
  "id": "TH-OC-50_0155",
  "url": "50155.jpg",
  "type": "page",
  "date": "21/6/2017"
 },
 "location": [
  {
   "id": "title_126",
   "polygon":
           [
            [343.9781962339, 283.4998933161],
            [1443.9524281467, 283.4998933161],
            [1443.9524281467, 506.5198093914],
            [343.9781962339, 506.5198093914]
           ],
   "type": "title"
  },
  {
   "id": "title_126_tl_0",
   "type": "textline",
   "polygon":
        [
            [344.0,292.0],
            [345.0,293.0],
            ...,
            [345.0,491.0],
            [344.0,492.0]
    ]
  }
  {...}
 ],
 "data": [
  {
           "value": "6 ème du jardinier de Sidon,Le diable à quatre avec des
               Divertissements",
           "id": "a_0",
           "type": "annotation",
           "location_id": "title_126"
  },
  {
           "value": "6 ème du jardinier de Sidon,Le diable à quatre avec des
               Divertissements",
           "id": "a_0_tl_0",
           "parent_id_list": "title_126",
           "type": "annotation",
           "location_id": "title_126_tl_0"
  },
  {
           "value": "6 ème du jardinier de Sidon,Le diable à quatre avec des
               Divertissements",
           "id": "a_0_tl_1",
           "parent_id_list": "title_126",
           "type": "annotation",
           "location_id": "title_126_tl_1"
  }
 ]
}
```

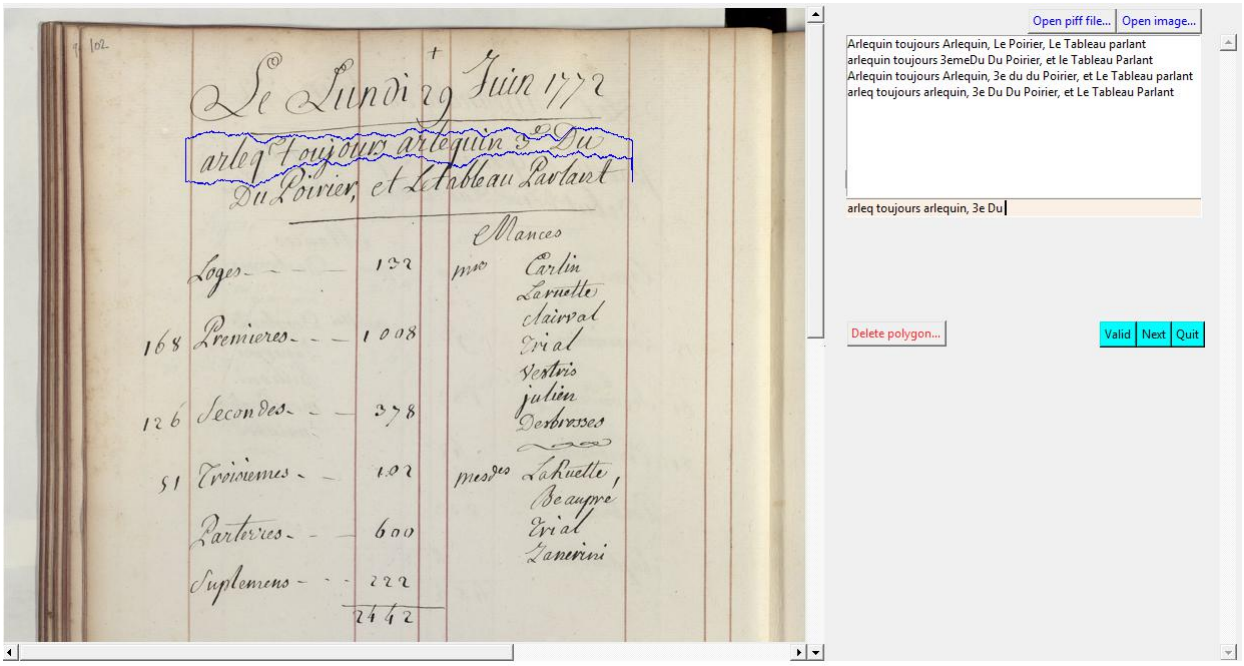


FIGURE 3.5 – Interface graphique permettant de vérifier manuellement les transcriptions des lignes de titre. La partie gauche montre les polygones et la zone sur la page globale. Le polygone bleu est le résultat de l'algorithme TLA. La partie droite fournit toutes les transcriptions candidates. L'utilisateur peut valider son choix, passer au polygone suivant (sans validation) ou supprimer un polygone avec ses transcriptions candidates.

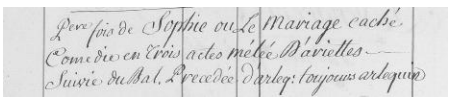
3.2.2 Bases de la Comédie-Italienne

Nous présentons dans cette section les deux bases d'images créées à partir des documents de la Comédie-Italienne. La première base est construite sur un unique registre TH-OC-53, tandis que la seconde, qui utilise les données de la plateforme participative, couvre un large éventail des registres disponibles.

Base sur le registre TH-OC-53

À l'aide de la méthode DMOS, nous avons tenté de détecter et segmenter automatiquement les zones de titre dans les registres. Cette méthode générique utilise un langage grammatical de descriptions créé pour cette tâche, avec un extracteur d'éléments terminaux basé sur les filtres de Kalman et un analyseur qui autorise la modification en cours d'étude pour s'adapter au document courant.

Une grammaire a été spécialement définie afin de détecter les filets présents dans les registres de la Comédie-Italienne qui, à certaines périodes, séparent les zones de date et de titre, et se trouvent dans les 25% du haut des pages. Cette étape permet d'obtenir des blocs de texte (Figure 3.6a). Ensuite, la séparation en ligne est réalisée grâce à la méthode *Seams Carving* (Arvanitopoulos et Süsstrunk 2014) comme nous l'avons déjà présenté précédemment à la section 3.2.1. Finalement, une correction d'inclinaison ainsi qu'une normalisation de la hauteur des lignes sont réalisées (voir Figures 3.6d et 3.6e).



(a) "Bloc original"

(b) "Lignes médianes"

(c) "Lignes séparatrices"

(d) "Ligne obtenue"

(e) "Ligne corrigée et recentrée"

Figure 3.6 – Étapes réalisées pour obtenir des lignes segmentées sur la Comédie-Italienne.

Nous avons constaté des erreurs engendrées à chaque étape du processus. Les méthodes de normalisation présentent des irrégularités sur les images finales. Suivant les registres ou les pages, du bruit s'ajoute sur les images, l'inclinaison est mal réalisée. La hauteur des majuscules, des hampes et des jambages est souvent responsable d'une mauvaise segmentation. Elles provoquent la segmentation d'une ligne en deux par exemple. Donc, parmi les lignes qui ont pu être segmentées et normalisées, nous avons opéré une sélection manuelle.

Finalement, 156 images de titre provenant du registre 53 ont été sélectionnées, car considérées comme idéales c'est-à-dire une ligne bien segmentée dont le(s) titre(s) ne sont pas coupés sur une autre ligne. Ces lignes ne contiennent pas uniquement des titres, mais également des mentions de jour de relâche.

Base multi-registres

Le système d'annotation assistée, décrit à la section 3.2.1, nous a permis de créer de nouvelles ressources d'images de lignes de titre étiquetées, et de formaliser les informations collectées dans un format dynamique. Grâce aux premières annotations collaboratives, de nouvelles segmentations de titres en bloc sont disponibles et assez fiables que nous pouvons traitées afin d'obtenir de nouvelles images de lignes de titres.

Pour cela, nous avons recueilli toutes les annotations relatives à la zone de titre réalisées sur la plateforme participative à la date du 30 juin 2017. Puis la détection et la segmentation en lignes ont été réalisées, comme décrites à la section 3.2.1. La partie la plus fastidieuse a été de vérifier et segmenter manuellement les transcriptions candidates associées initialement aux zones de titres contenant plusieurs lignes. Pour rappel, la principale directive qui a été faite aux contributeurs pour les aider dans leurs tâches était d'utiliser une transcription diplomatique "souple". Cela implique de ne pas mettre toutes les majuscules présentes dans les titres, transcrire en entier les termes qui ont été abrégés ainsi que moderniser, sans abréviation, les termes comme les dates où il est possible de trouver "7^{bre}" à transcrire en "Septembre". Ces directives concordent avec l'objectif de RECITAL, mais soulèvent des problèmes et questions pour utiliser les transcriptions candidates pour de la transcription automatique. En juin lorsque nous avons récupéré les informations liées aux zones de titres, il y avait encore assez peu de contributeurs, ce qui implique peu de propositions pour une même zone pour arriver à un consensus. Sur l'ensemble des zones, seulement 4,3 % convergeaient vers un consensus. De plus, parmi les 926 transcriptions candidates de titres, nous avons comptabilisé environ 40 % d'annotations par un membre de notre équipe, 30 % par des anonymes et le reste des contributeurs connus ont annoté moins de 10 % . Finalement, une seule

personne a réalisé la phase de post-annotation (moi). Ce travail a principalement consisté à remettre les majuscules et la ponctuation manquantes ou faire un consensus entre les différentes propositions par exemple {Julie La Clochette ; Julie, La clochette ; Julie, la clochette} pour “Julie, La Clochette” ; remettre ou corriger les abréviations, par exemple avec “86” proposé au lieu de “etc” ; et vérifier que les termes n’avaient pas été traduits comme “extravagante” qui avait été proposé au lieu “Stravagante”. Les cas où un utilisateur ne proposait pas de transcription pour une zone n’ont pas posé de problème lors de la validation des lignes, car il y avait dans chacun de ces cas une ou plusieurs propositions faites par d’autres contributeurs. Pour finir, il est difficile de fournir un accord inter-annotateur suite à cette annotation car les transcriptions candidates ont dû être coupées suite à la segmentation des zones en lignes et une seule personne a opéré la validation de ces transcriptions.

Finalement, 971 lignes de titre et leurs transcriptions ont pu être sélectionnées et validées avec l’outil décrit précédemment (voir section 3.2.1). En dehors de la normalisation en hauteur des images, nous n’avons pas réalisé d’autres traitements comme la correction de l’inclinaison des lignes. Nous présentons 6 exemples provenant de 6 registres différents à la figure 3.8. Cet échantillon montre la diversité observée entre les registres en termes de papiers, d’encre, de style, de longueur et de langue (l’image 3.8a est en italien). La répartition du nombre d’images de titre collectées en fonction des registres est présentée à la figure 3.7.

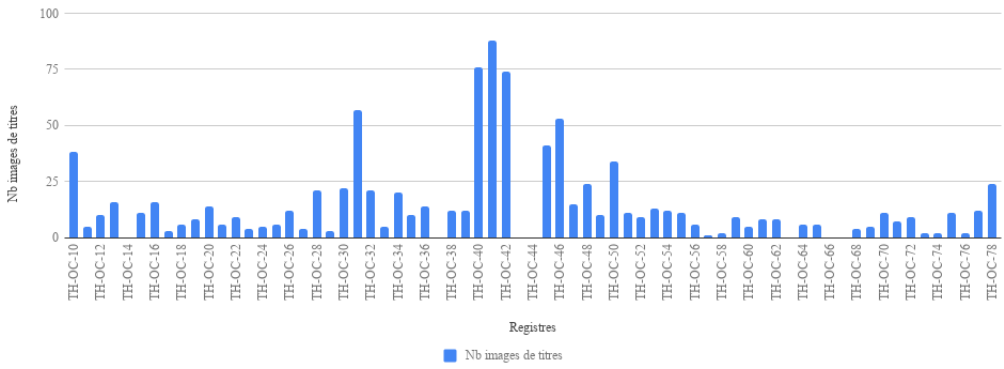


FIGURE 3.7 – Répartition des images de titres finales suivant les registres

3.3 Autres Ressources mobilisées

Les ressources annotées disponibles du XVIII^{e} sont assez rares et celles en français inexistantes à notre connaissance. C’est pour cette raison que nous nous

(a) "Furbarie di Scapino" (TH-OC-10)

(b) "Arlequin maitre d amour" (TH-OC-30)

(c) "arlequin toujours arlequin" (TH-OC-34)

(d) "arlequin Persécuté par la dame jnvisible" (TH-OC-39)

(e) "Les aveux jndiscrits, arlequin cru mort" (TH-OC-45)

(f) "La Melomanie" (TH-OC-78)

FIGURE 3.8 – Exemples d'images de lignes de titres de RCI associées à leur transcription.

sommes tournés vers un ensemble de ressources couvrant un large spectre de caractéristiques liées aux images et aux langages.

3.3.1 Georges Washington

La base de données Georges Washington (RATH et al. 2002) est constituée de 20 pages de correspondance en anglais du président éponyme, avec deux scripteurs connus, mais dont les écritures sont similaires. Un exemple d'une page est fourni en figure 3.10a.

La base fournit une version des pages originales en niveaux de gris avec les coordonnées des mots, ainsi qu'une version segmentée et normalisée des images de lignes (656 images) et de mots (4 894). Les traitements appliqués sur les images sont une normalisation de la hauteur des images à 120 pixels, une binarisation et une correction de l'inclinaison. 4 répartitions pour faire de la validation croisée sont également proposées, avec 10 pages pour l'apprentissage, 5 pour la validation et 5 pour le test.

Nous avons utilisé la première validation proposée pour les expérimentations avec une version non normalisée des images de lignes et de mots.

3.3.2 Les Esposalles

La base de données Les Esposalles (Romero et al. 2013) a été créée à partir de 291 livres de registres de mariages du XV^{e} au XVII^{e} siècle provenant des archives de la cathédrale de Barcelone. La base que nous utilisons "LICENSES MARIAGES" est un sous-ensemble de ces documents qui est composé de 174 pages d'un livre couvrant 1617 à 1619 en catalan. Un exemple de ces pages est donné à la figure 3.10b : en haut de la page se trouve le mois ainsi que l'année ; le jour est mentionné dans la colonne de gauche, s'il est différent du mariage précédent, avec le nom du marié ; la zone centrale contient, pour chaque mariage, les informations relatives aux deux époux comme leurs noms, leurs statuts, leurs métiers ainsi que leur lieu de vie.

Les pages entières fournies permettent d'extraire 56 378 mots avec leur transcription. Il y a peu de variations dans le style de l'écriture donc il est possible de considérer qu'il y a un unique scripteur. Les transcriptions ont été faites manuellement avec un protocole bien défini par un expert. La notation pour les abréviations, incluant des exposants comme sur la figure 3.9, est mentionnée par "^(texte)". Nous avons fait le choix de supprimer les parenthèses et l'accent circonflexe dans le vocabulaire ainsi que dans les transcriptions. Les images de cette base ont l'avantage d'avoir un fond marqué par le temps, c'est-à-dire bruité comme il est possible de le voir sur la figure 3.10b, comme cela est le cas pour les registres de la Comédie-Italienne.

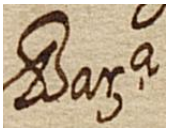


Figure 3.9 – Exemple d'abréviation contenue dans la base Les Esposalles : "Barcelona" transcrit par "Barc^(a)"

La taille du vocabulaire est très faible par rapport aux nombres d'images de mots. Nous avons 60 000 images pour seulement 4 500 mots distincts. Cette forte baisse est facile à expliquer, en effet, tous les enregistrements de mariage sont rédigés suivant un modèle. Le terme le plus représenté est l'article "de" avec 11 366 occurrences, suivi par les termes "dia", "doncella", "filla" et "reberem" avec 1 500 occurrences chacun.

3.3.3 RIMES

La base de données RIMES (Reconnaissance et Indexation de données Manuscrites et de fac similÉS / Recognition and Indexing of handwritten documents and faxes) (Grosicki et El-Abed 2011) est composée de différents documents administratifs français, comme des fax, des formulaires ou encore des courriers datant de 2008. 1 300 volontaires ont répondu librement avec leur propre vocabulaire à 1 des 5 scénarios parmi les 9 thèmes qui leur étaient proposés. L'identité utilisée devait être fictive tout en conservant le sexe réel. La seule contrainte imposée était l'utilisation d'un papier blanc sans ligne directrice et d'un stylo noir. Un exemple de ces documents est fourni à la figure 3.10c.

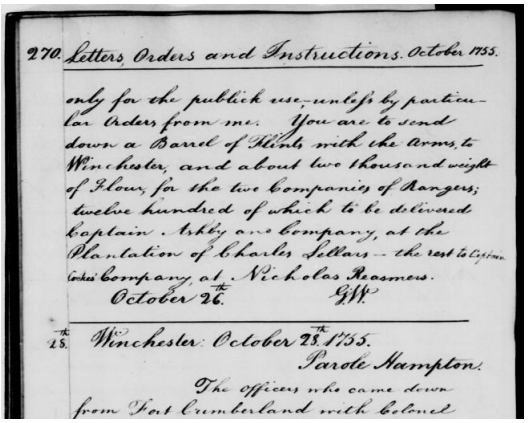
Cette collecte a permis de constituer une base avec 12 723 pages. Pour la compétition ICDAR 2011, une répartition des données a été fournie avec 12 111 lignes (dont 11 333 pour l'apprentissage) et 66 979 mots (dont 51 739 pour l'apprentissage). Ces images sont fournies en niveaux de gris avec une taille moyenne de 2000 pixels sur 150.

Le vocabulaire et son orthographe sont modernes, centrés autour de l'administration avec des entités nommées (Noms, Villes, entreprises), des adresses email et postales, des numéros clients. Pour pouvoir utiliser cette base, nous avons fait le choix d'écarter tous les signes contemporains comme les adresses email, les numéros de téléphone ou encore les identifiants incluant le symbole "#". Le vocabulaire obtenu est formé de plus de 6 000 mots différents.

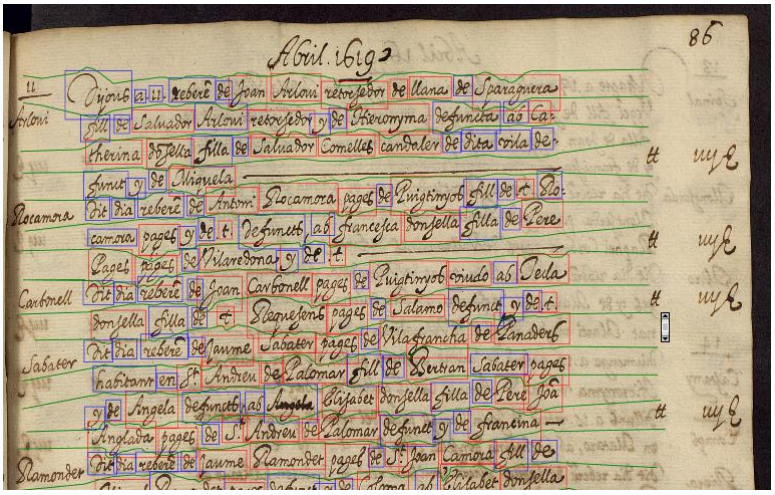
3.3.4 Google Livres sur la Comédie-Italienne

Le vocabulaire utilisé dans les bases d'images présentées précédemment étant fermé et dédié à un domaine à chaque fois, nous avons cherché à enrichir les bases utilisées avec des ressources linguistiques traitant de la Comédie-Italienne. Nous avons cherché, dans un premier temps, des documents produits par des contributeurs du projet CIRESFI comme "Le répertoire de la Comédie-Italienne de Paris (1716-1762)" d'Emanuele de Luca ou encore des pages de Wikipédia relatives à certains auteurs connus. Cependant, le problème majeur de ces documents est qu'ils sont normalisés en effet, le vocabulaire et l'orthographe sont modernes, et les titres sont souvent écrits dans leur version longue. C'est pour cela que nous nous sommes tournés dans un second temps vers des documents disponibles en grand nombre du XVIII^{e} siècle, mais souvent sous-exploités.

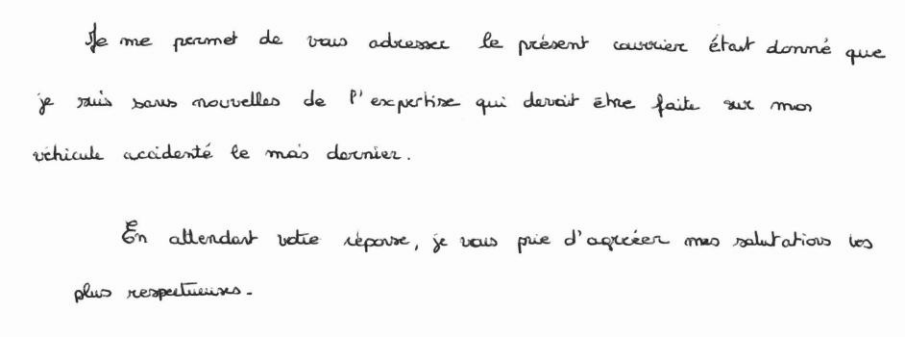
En effet, en cherchant des ressources potentielles en français qui avait pour thème la Comédie-Italienne, nous avons constaté que sur Google Livres, 419 000



(a) “Georges Washington”



(b) “Les Esposalles”



(c) “Rimes”

FIGURE 3.10 – Exemples des documents contenus dans chaque base de données sélectionnée

livres français faisant référence à la Comédie-Italienne étaient disponibles. Parmi l'ensemble de ces œuvres, nous avons identifié des scripts bilingues (en Italien et en Français), des répertoires d'œuvres, des livres d'anecdotes sur le théâtre italien (c'est-à-dire incluant les théâtres forains) et portant sur certaines saisons, des prologues d'ouvertures de saison ainsi que des compliments de clôture...

Finalement, nous avons sélectionné 23 ouvrages qui sont détaillés dans la table 3.2 en précisant leur année de publication, le titre de l'œuvre, son type. Ils sont tous en français classique qui diffère du français moderne par l'orthographe, la syntaxe et le vocabulaire utilisés. Les douze ouvrages contenant des scripts d'œuvre, ils permettent de fournir un vocabulaire propre aux comédies de cette époque, car les termes utilisés n'étaient pas forcément ceux de la vie courante, ce qui contraste avec les ouvrages d'anecdotes.

Les fichiers au format PDF sont disponibles et incluent une version texte du contenu grâce à une océrisation des images par Google.

L'avantage linguistique principal qu'offre cette nouvelle ressource porte sur le fait que le texte n'a subi aucun traitement supplémentaire pour normaliser le vocabulaire. L'orthographe utilisée est celle de l'époque. Par exemple, les termes au singulier finissant par "ment" deviennent "mens" au pluriel, le "t" final se transforme en "s". La forme longue du "s" est également présente.

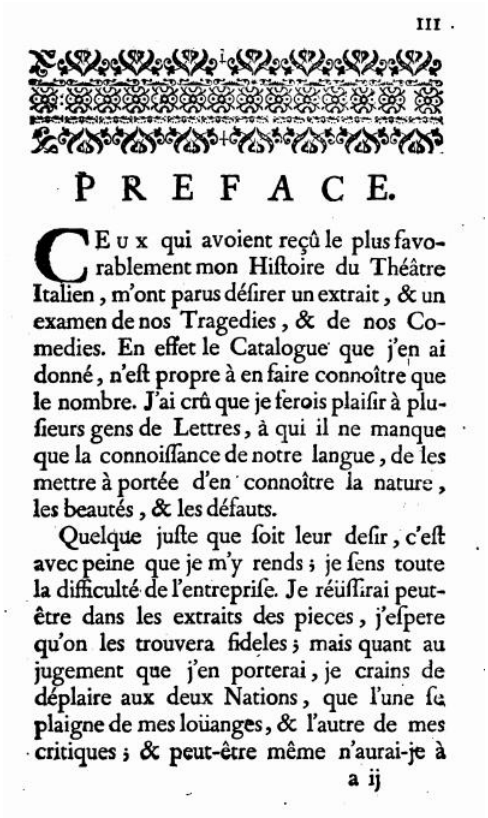
Bien que nous ne souhaitons pas normaliser le texte, il est nécessaire de faire des pré-traitements afin de nettoyer le texte obtenu par le OCR. En effet, l'outil de Google a pris une décision pour chaque tache sur le papier, mais également pour chaque dessin ou frise qu'il considère comme une séquence de plus de 20 caractères non interrompue.

Un exemple du texte obtenu par le OCR est fourni dans la table 3.3. La qualité des documents et le style de l'écriture utilisé provoquent des difficultés au moment de l'extraction du texte. Par exemple, "eſt" n'est jamais bien reconnu dans cet exemple (identifié en rouge dans le résultat obtenu). Des caractères de ponctuation sont souvent ajoutés pour correspondre avec des tâches liées au temps sur la page (voir caractères rouges). C'est pour cela qu'au moment de construire le vocabulaire, nous avons filtré les séquences avec une taille supérieure à 15 caractères. Puis, nous conservons les termes apparaissant plus de 5 fois dans l'ensemble des ouvrages sélectionnés pour tenter de supprimer les substitutions de caractères non fréquentes. Toutes ces erreurs de transcription sur un document tapuscrit historique mettent clairement en avant les difficultés restantes pour désambiguïser le fond du contenu, mais également les formes des caractères.

Au total, 29 526 mots constituent le vocabulaire des Google Livre de la Comédie-

TABLE 3.2 – Ensembles des 23 œuvres de la Comédie-Italienne sélectionnées.

Date	Titre du livre	Type	Pages
1654	Les Nopces de Pelee et de Thetis	Script de la pièce	47
1726	Arlequin Toujours Arlequin	Script de la pièce	33
1729	Le Nouveau Théâtre Italien ou Recueil General des Comédies T. 3	Recueil de scripts (Français et Italien) Italien	473
1731	Histoire du Théâtre Italien depuis la Décadence T. 2	Résumé de pièces ordonnées par type	417
1732	Le Jaloux	Script de la pièce	143
1733	L'isle du Divorce, comédie	Script de la pièce	43
1737	Complimens pour la Closture et pour l'Ouverture	Compliments	9
1753	Le Nouveau Théâtre Italien ou Recueil General des Comédies T. 1	Recueil ordonnée par date de représentation	495
1760	Catalogue des Livres de la Bibliothèque de Feu M. G***	Bibliographie ordonnée par type d'ouvrage	793
1762	Le Fils d'Arlequin Perdu et Retrouvé	Script de la pièce	17
1763	Bibliographie Instructive ou Traité de la Connaissance des Livres Rares et Singuliers	Bibliographie ordonnée par type d'ouvrage	703
1767	Compliment pour la Clôture de la Comédie Italienne	Compliments	23
1767	Prologue d'Ouverture pour la Comédie Italienne	Prologue	29
1769	Histoire Anecdotique et Raisonnée du Théâtre Italien T. 3	Extraits de pièces Italiennes et Françaises	551
1775	Anecdotes dramatiques T. 3	Anecdotes ordonnées par noms des auteurs et acteurs	593
1777	Les Trois Théâtres de Paris ou Abrégé Historique de l'établissement de la Comédie Française, comédie Italienne et de l'Opéra	Historique et anecdotes	323
1778	Les Trois Jumeaux Vénitiens	Script de la pièce	69
1783	Le Nouveau Théâtre Italien ou Recueil General des Comédies	Recueil ordonnée par date de représentation	569
1786	De l'Art de la Comédie de l'Imitation, nouvelle édition, T.2	Extraits de pièces et anecdotes	445
1786	Le Valet Rusé, ou Arlequin Muet	Script de la pièce	39
1788	Annales du Théâtre Italien, depuis son origine jusqu'à ce jour	Anecdotes ordonnées par années	673
1789	La Fausse Magie, Comédie	Script de la pièce	37
1820	Œuvres Complètes de Regnard T. 5	Recueil	451



III .

PREFACE.

CEux qui avoient reçû le plus favorablement mon Hiſtoire du Théâtre Italien, m'ont parus déſirer un extrait, & un examen de nos Tragedies, & de nos Comedies. En effet le Catalogue que j'en ai donné, n'eſt propre à en faire connoître que le nombre. J'ai crû que je ferois plaiſir à pluſieurs gens de Lettres, à qui il ne manque que la connoiſſance de notre langue, de les mettre à portée d'en connoître la nature, les beautés, & les défauts.

Quelque juſte que ſoit leur déſir, c'eſt avec peine que je m'y rends; je ſens toute la difficulté de l'entrepriſe. Je réüſſirai peut-être dans les extraits des pieces, j'eſpere qu'on les trouvera fideles; mais quant au jugement que j'en porterai, je crains de déplaire aux deux Nations, que l'une ſe plaigne de mes loüanges, & l'autre de mes critiques; & peut-être même n'aurai-je à

a ij

’ï
III .
EWÆWNWŒQDÇWÊNÆNQP WWWÊMWŒWË-
SÕWÆÏWÆWEWAN “1*- f? ? 1’213 7—-— -Tcy-
?POTS PW ? ? 3 ;’ –Wd- v : ..fr ï rf-’J QR

PREFACE.

ØEUx qui avoient reçû le plus favo
rablement mon Hiſtoire du Théâtre
Italien , m’ont parus déſirer un extrait , 8c un
examen de nos Tragedies , 8c de nos Co
medies. En effet le Catalogué que j’en ai
donné , n’e propre à en faire con-noître que
le nombre. J’ai crû que je ferois plaiſir à plu ſieurs
gens de Lettres, à qui il ne manque
que la connoiſſance de notre langue , de ’les
mettre à portée d’en ’ connoître la nature ,
les beautés , & les défauts.

Quelqüe juſ’ce que ſoit leur déſir , c’ e
avec peine que je m’y rends ; je ſens toute
__ la diſſculté de l’entrepriſe. Je réüſſrai peut
être dan-S les extraits des pieces , j’eſpere
qu’on les trouvera fidelesz mais quant au
jugement que j’en porterai, je crains de
déplaiſe aux deux Nations, que l’une ſe
plaigne de mes loüanges, 8c l’autre de mes
-critiquess 8C peut-être même n’aurai-je à
a 1)

TABLE 3.3 – Exemple du texte extrait par l’outil OCR sur une page de “Complimens pour la Closture”. En rouge, les erreurs liées aux imperfections du document ; en vert, les erreurs sur le mot “eſt” ; en bleu, erreurs sur d’autres caractères.

Italienne (GCI) partagé entre un ensemble d’apprentissage (26 573) et validation (2 953).

3.3.5 Synthèse en comparaison avec les registres de la Comédie-Italienne

Toutes les informations que nous avons pu lister sur l’ensemble des ressources images et linguistiques sont résumées dans la table 3.4.

Chaque ressource de type *Image* sélectionnée possède au moins une caractéristique commune avec les registres de la Comédie-Italienne. Il nous semblait

important de sélectionner des ressources historiques incluant les mêmes particularités stylistiques que dans nos registres comme le “f” et avec des imperfections similaires au niveau du papier.

L'ensemble des ressources possèdent des caractéristiques différentes (période, langage) ce qui aura un impact sur les systèmes et leur apprentissage tout comme la quantité d'images disponibles. En effet, cela influera sur le temps d'exécution, les performances de reconnaissance et la réalisation de l'apprentissage par transfert.

La diversité graphique et stylistique des images contenues dans les bases de données est un atout pour notre étude d'apprentissage par transfert. Les images sont conservées dans l'état, avec des lignes courbées, des hampes et des jambages empiétant sur les lignes voisines, des taches sur le papier...

Le vocabulaire de chaque ressource est dédié à un domaine dans une langue spécifique. Il faut noter que pour toutes les bases de données, les fautes d'orthographe ont été conservées dans les transcriptions. GCI est la ressource la plus proche des registres de la Comédie-Italienne, c'est pour cette raison que nous l'avons constituée. Cependant, elle n'est que linguistique et ne pourra pas être utilisée pour la tâche de reconnaissance d'écriture des registres. La ressource Les Esposalles a l'avantage d'être constituée majoritairement d'entités nommées comme les registres de la Comédie-Italienne. Nous constatons également que les ressources RIMES et Les Esposalles sont les plus fournies en images de mots, cependant leur vocabulaire semble plus limité que ce qu'il pouvait paraître. Nous avons évalué que pour les ensembles d'images de RIMES et Les Esposalles , seulement 5 à 10% des mots sont différents. Par exemple, RIMES possède 51 739 images de mots, mais 4 477 mots distincts y sont représentés.

3.4 Conclusion

Dans ce chapitre 3, nous avons présenté l'ensemble des ressources utilisées pour les expériences à venir. Pour étudier l'apprentissage par transfert de connaissances, un des facteurs clés est le choix des ressources en fonction des données cibles.

Dans un premier temps, nous avons détaillé le fonctionnement global de la plateforme RECITAL qui a pour but d'annoter les registres de la Comédie-Italienne avec la participation de volontaires. Associée à un système de segmentation et vérification manuelle, nous avons pu créer une base de données d'images de lignes couvrant un grand nombre de registres différents.

Les deux bases images construites sur la Comédie-Italienne ont une quantité

TABLE 3.4 – Taille du vocabulaire et distribution des images de mots et lignes pour chaque ressource. Les points communs entre les ressources et les registres RCI sont mis en surbrillance et indiqués par **.

Distrib.		GCI	Georges Washington	RIMES	Les Esposalles	Wiki	RCI-53	RCI-IAC
Type		Linguist.	Images	Images	Images	Linguist.	Images	
Langage		Français **	Anglais	Français **	Catalan	Français **	Français (et Italien)	
Période		XVIII^e **	XVIII^e **	XXI^e	XVIII^e **	XXI^e	XVIII^e	
Vocabulaire		Comédie Italienne **	Correspondance	Administratif	Registres Mariage	Général	Titres pièces	
Images Mots	Apprentissage		2 402	51 739	45 102			
	Validation		1 199	7 464	5 637			
	Test		1 292	7 776	5 637			
Images Lignes	Apprentissage		325	11 333				582
	Validation		168	1 332				195
	Test		163	778			156	194
Vocab.	Apprentissage	26 573	660	4 477	2 565	24 456	0	0
	Validation	2 953	521	1 578	629	3 843	0	0
	Test	0	431	629	1 627	1 928		1 431

d'images trop faible pour être utilisées comme unique ressource d'apprentissage pour des systèmes de reconnaissance. C'est pour cette raison que nous avons sélectionné trois autres ressources images et constitué une nouvelle ressource linguistique autour de la Comédie-Italienne.

Les travaux effectués sur la création de la plateforme d'annotation participative ainsi que la constitution d'une base de données sur les images de lignes de la Comédie-Italienne ont été valorisés dans une publication réalisée dans la conférence LREC (Granet et al. 2018a).

Lors des expérimentations des deux systèmes décrits dans les prochains chapitres, aucun autre traitement en dehors de ceux décrits dans ce chapitre n'a été réalisé. Cela signifie que la base multi-registre a seulement été normalisée en hauteur. Tandis que la base sur le registre TH-OC-53 a en plus vu l'inclinaison de ces lignes corrigées. Notre premier système *end-to-end* est présenté dans le chapitre 4 suivant.

4

Système *end-to-end* avec apprentissage par transfert de connaissances

Sommaire

Introduction

Dans le chapitre 2, nous avons présenté les systèmes standards de reconnaissance d'écriture à travers les différentes étapes qui les composent : les pré-traitements des

documents et l'extraction des caractéristiques des images à traiter, les différentes approches et leur fonctionnement ainsi que les méthodes du traitement du langage qui peuvent améliorer les performances. Les modèles les plus performants sont des réseaux de neurones multi-dimensionnels tirant parti du meilleur des méthodes existantes actuelles : modèle d'attention, modèle récurrent, fonction de coût CTC. Cependant, leurs architectures complexes présentent deux points délicats. Le premier est la quantité de données nécessaires pour réaliser leur apprentissage. Le second est la spécialisation potentielle du système. Dans le chapitre précédent, nous avons présenté les documents de la Comédie-Italienne sur lesquels nous travaillons. Au début de cette étude, nos connaissances de ce corpus étaient limitées, voire inexistantes. Il était donc impossible de réaliser directement des réseaux aussi complexes que ceux de l'état-de-l'art pour répondre à notre étude.

Pour répondre à nos besoins, nous nous intéressons particulièrement à l'apprentissage transductif par transfert de connaissances qui se concentre sur l'adaptation de domaine (Pan et Yang 2010a) en utilisant diverses ressources sources pour apprendre un système et l'appliquer sur une ressource cible différente. Nous souhaitons expérimenter cette méthode et l'approfondir en multipliant le nombre de données annotées utilisées comme source et en ajoutant un transfert des poids des paramètres appris sur un ensemble de ressources et utilisés pour une ressource cible différente. À notre connaissance, peu d'études s'intéressent à l'apprentissage par transfert de connaissances pour la reconnaissance d'écriture manuscrite, et qui plus est, sur des documents anciens. Dans un premier temps, nous avons besoin de concevoir un premier système simple, se basant sur des composants ayant fait leurs preuves dans le domaine. Une fois que ce réseau aura atteint des résultats proches de l'état-de-l'art, nous pourrons le tester avec l'apprentissage par transfert de connaissances.

Dans ce chapitre, nous présentons le premier système de reconnaissance que nous avons expérimenté, à partir d'un réseau BLSTM-CTC. Dans un premier temps, nous présentons le choix des méthodes utilisées pour l'extraction de caractéristiques. Dans un second temps, nous nous attarderons sur les paramètres du réseau BLSTM, la fonction de coût CTC qui lui est associée ainsi que les méthodes de décodage utilisées.

4.1 Caractéristiques extraites

Pour le système de reconnaissance d'écriture, une fois les images découpées comme il se doit, il faut extraire un ensemble de caractéristiques pour les représenter et les donner en entrée du système qui les interprétera. Nous avions présenté dans

la section 2, les différentes méthodes d'extraction de caractéristiques suivant deux grandes familles : structurelle et directionnelle. Nous expérimentons trois méthodes différentes pour définir les caractéristiques : en utilisant les pixels directement, en construisant l'histogramme des gradients orientés et avec un réseau entièrement à convolution.

4.1.1 Méthode basée sur les pixels

La première méthode considérée pour l'extraction de caractéristiques se base sur la position des pixels. Parmi l'ensemble des estimations de caractéristiques possibles, nous avons sélectionné les caractéristiques statistiques et géométriques suivantes :

— hauteurs des profils haut et bas (2 informations) ;
— le nombre de transitions entre l'encre et le fond (1 information) ;
— la somme des pixels normalisée (1 information).

Nous obtenons donc 4 caractéristiques basées sur des informations locales. Nous utilisons une fenêtre d'observation de un pixel. Cela permet de résumer les connaissances de l'image à l'essentiel indépendamment de sa longueur. Dans le cadre de nos expériences, nous utilisons cette méthode uniquement avec une version de la ressource Georges Washington normalisée et distribuée. La hauteur des images a été fixée à 120 pixels et les images ont été binarisées.

4.1.2 Histogramme des gradients orientés

La seconde méthode expérimentée est la méthode des histogrammes des gradients orientés (HOG). Cette méthode donne une information différente de la précédente puisqu'elle se base sur la forme du tracé. Pour appliquer cette méthode, il nous faut définir une taille de fenêtre ainsi que le pas d'exécution. Nous nous basons sur l'ensemble des études qui ont déjà été menées pour définir la meilleure fenêtre parmi celles proposées par (Terasawa et Tanaka 2009 ; Toledo et al. 2016). Finalement, le choix a été fait de n'utiliser qu'une fenêtre glissante de 20 pixels de large avec un pas de 2. Les ressources utilisées ont une hauteur normalisée à 120 pixels. Chaque fenêtre est divisée en 2×4 sous-fenêtres dans lesquelles la méthode HOG est appliquée selon 8 orientations possibles. Nous obtenons ainsi 8 histogrammes, ce qui donne 64 caractéristiques pour chaque instant comme la figure 4.1.

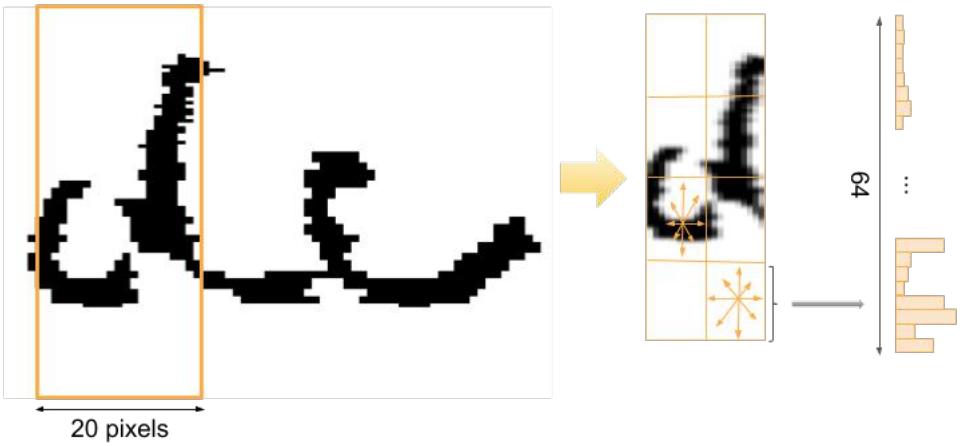


FIGURE 4.1 – Exemple de segmentation avec une fenêtre de 20 pixels de large, divisé en 8 portions dans lesquelles un histogramme selon 8 directions est calculé. Finalement, l'ensemble des 8 histogrammes sont concaténés pour former un unique vecteur de 64 caractéristiques.

4.1.3 Réseau de neurones à convolution

La troisième méthode expérimentée est un réseau de neurones entièrement à convolution (RNC), pour extraire les caractéristiques, qui peut être intégré directement dans le système de reconnaissance d'écriture choisi. Il nous faut définir le nombre de convolutions, avec le nombre de filtres ainsi que la taille du noyau. Cependant, il n'existe pas une unique configuration pour définir le nombre de filtres, et le nombre de couches à utiliser pour réaliser ce système : cela change suivant le type des images ou de la tâche considérée. Un des premiers systèmes performants et un des plus simples pour réaliser de la classification est *Alexnet* (KRIZHEVSKY et al. 2012), composé principalement de 5 convolutions avec des tailles de filtres de 96 à 256 (avec un maximum de 384). La complexité des réseaux a rapidement explosé avec l'augmentation continue du nombre de couches à convolution utilisées. Par exemple, VGG-19 (SIMONYAN et ZISSERMAN 2014a) avec 14 convolutions (HE et al. 2016) est le réseau le plus profond expérimenté sur ImageNet avec 32 couches, mais le nombre de filtres utilisés est plus linéaire en commençant à 64 à chaque fois et doublant à chaque nouveau bloc.

En nous basant sur les architectures existantes, nous avons expérimenté deux modèles différents pour extraire les caractéristiques. La figure 4.2 donne les représentations graphiques de chacune d'elles avec le détail de l'architecture dans la table 4.1. Les deux réseaux à convolution utilisent la même entrée qui correspond à une image avec une hauteur fixe de 120 pixels et une largeur variable notée t. La

partie commune des deux réseaux est composée de trois couches de convolution avec un noyau de taille 5×5, correspondant à la partie grisée de la figure 4.2. Pour la seconde partie de chaque réseau, nous avons souhaité comparer les effets du nombre de filtres sur les résultats de la reconnaissance de caractères. Pour cela, nous avons défini un premier réseau, appelé *RNC32*, avec 3 couches de convolution et uniquement 32 filtres. Le second réseau, appelé *RNC128*, possède 3 couches de convolution avec un nombre de filtres qui double : 64 et 128 filtres. Chacune des couches convolutionnelles est suivie par une couche de sous-échantillonnage, appelée *Max-Pooling*. Les caractéristiques obtenues en sortie de chaque réseau ont pour dimension $32\times1\times\frac{t}{20}$ pour *RNC32* et $128\times1\times\frac{t}{20}$ pour *RNC128*. Ce réseau est connecté directement en amont du réseau de reconnaissance d'écriture pour réaliser les apprentissages des 2 réseaux, simultanément.

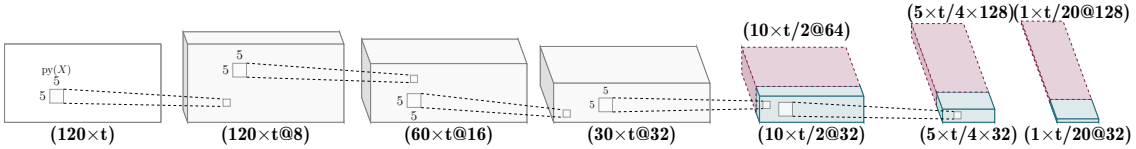


FIGURE 4.2 – Représentation graphique des deux architectures de la partie de réseau de neurones à convolution utilisées pour l'extraction de caractéristiques. La version bleue correspond au réseau *RNC32* et la version rose correspond au réseau *RNC128*.

4.2 Modèle BLSTM-CTC

Les caractéristiques extraites sur les images à partir des différentes méthodes (locale, HOG ou RNC) alimentent le modèle de reconnaissance d'écriture manuscrite. Nous avons introduit précédemment les réseaux de neurones de type *Long Short Term Memory* (LSTM) dans le chapitre 2.2.2. Ces dernières années, bien que plusieurs systèmes de type multi-dimensionel aient supplanté les autres systèmes (GROSICKI et EL ABED 2009b ; MOYSSET et al. 2014), nous nous concentrons sur un simple réseau Bidirectionel-LSTM associé au CTC.

L'architecture du réseau est celle présentée à l'origine dans (GRAVES et al. 2009 ; GRAVES 2012b). Cela signifie que, initialement, les deux couches cachées du réseau *forward* et *backward* possèdent 100 cellules LSTM chacune. Pour d'autres expériences, nous avons expérimenté un autre réseau avec seulement 50 cellules dans chaque couche pour observer l'impact réel de la taille de ces couches cachées sur les résultats. En amont, la couche d'entrée dépend du nombre de caractéristiques extraites à chaque instant : elle varie dans nos expériences de 4 (caractéristiques

TABLE 4.1 – Architecture du réseau de neurones entièrement à convolution pour l'extraction de caractéristiques. Les noyaux des couches utilisés sont au format (noyau)@ nb filtres.

Nom	**Type de couche**	**Taille du fitre**	
		RNC32	*RNC128*
Image	Image d'entrée	120×1×t	
Conv 1	Convolution	(5×5)@8	
Maxpool 1	MaxPooling	(2×1)	
Conv 2	Convolution	(5×5)@16	
Maxpool 2	MaxPooling	(2×1)	
Conv 3	Convolution	(5×5)@32	
Maxpool 3	MaxPooling	(3×2)	
Conv 4	Convolution	(5×5)@32	(5×5)@64
Maxpool 4	MaxPooling	(2×2)	
Conv 5	Convolution	(5×5)@32	(5×5)@128
Maxpool 5	MaxPooling	(5×5)	
Sortie		(1×t/20)@32	(1×t/20)@128

locales), 64 (HOG), 32 ou 128 (RNC) neurones et, dans le cadre du RNC, le réseau est directement branché sur l'entrée des couches cachées. Ensuite, la somme pondérée des deux couches récurrentes cachées est fournie à la couche de sortie construite avec 75 neurones et utilisant la fonction d'activation *softmax*. L'ensemble de ces neurones correspond aux 52 lettres minuscules et majuscules, aux 10 chiffres, et aux signes de ponctuation [,. ; : ! ? -ç&_]. Le dernier neurone de sortie (75ème) correspond au symbole *blank* qui permet au système de ne pas se prononcer à chaque instant. La fonction de coût CTC est utilisée pendant l'apprentissage afin de construire le treillis incluant l'ensemble des chemins possibles pour une séquence donnée. Le détail du fonctionnement du CTC est décrit en 2.2.2. La sortie fournie peut être décodée par plusieurs algorithmes, tels que l'algorithme de transfert de jeton ou le décodage de recherche par préfixe (GRAVES 2012b) pouvant inclure un modèle de langage. Nous utilisons la méthode du meilleur chemin pour nos expériences : à chaque pas de temps, le nœud le plus actif est sélectionné, ce qui donne finalement le chemin le plus probable.

4.3 Résultats et observations

Dans cette section, nous présentons l'ensemble des expérimentations utilisant le modèle BLSTM-CTC et les différentes caractéristiques définies. Les premières expérimentations visent à évaluer si le modèle, aussi simple soit-il, est capable de décoder correctement les séquences dans le cadre d'un apprentissage classique. Puis, nous présentons les expériences mettant en œuvre l'apprentissage par transfert de connaissances.

4.3.1 Méthode d'évaluation des performances

Pour prévenir le sur-apprentissage du système, l'apprentissage du réseau est arrêté dès que le coût calculé sur l'ensemble de validation ne diminue plus après 5 itérations. Cette méthode est appelée *early stopping*.

Pour évaluer les performances de notre système, nous avons utilisé le taux de reconnaissance au niveau des caractères (TRC) et au niveau des mots (TRM). Le TRC se base sur la distance d'édition (ou distance de Levenshtein) normalisée, définie par

$$TRC = \frac{N - (S + D + I)}{N} \times 100,$$

où N correspond au nombre de caractères dans les images de mots références, S au nombre de substitutions de caractères dans la séquence, D au nombre de suppressions de caractères et I au nombre d'insertions de caractères. Le TRM_L est calculé de la même manière avec le nombre de mots corrects dans une séquence de type. Pour les résultats sur des images de mots, le TRM est une réponse binaire, la réponse est correcte ou non. Ces deux mesures sont sensibles à la casse et considèrent l'espace comme un caractère dans le cadre de l'évaluation des lignes. Pour l'étape de décodage, aucun modèle de langage ou dictionnaire n'est utilisé sur l'ensemble des expériences. À travers ces expériences, nous visons à définir un système simple capable d'effectuer l'apprentissage par transfert de connaissances.

4.3.2 Données source et cible de domaine identique

4.3.2.1 Premiers résultats et observations

Les résultats d'expériences similaires aux nôtres mettant en œuvre des BLSTM-CTC sont présentés dans la partie haute du tableau 4.2, pour fournir une indication des résultats à atteindre même si nous n'utilisons pas les mêmes ressources. Ces

études utilisent des ressources beaucoup plus fournies que sont IAM (Marti et Bunke 2002) [1] et RIMES : la base IAM possède 20 fois plus de lignes que Georges Washington et RIMES contient le double de mots. Cela leur permet de proposer des évaluations directement sur les lignes, tandis que nous devons opérer une étape préliminaire d'apprentissage à partir des mots. Dans la partie centrale du tableau 4.2, les résultats des expérimentations sur la reconnaissance des mots de Georges Washington (GW_M) sont présentés. Finalement, la dernière partie du tableau 4.2 correspond aux résultats obtenus sur la reconnaissance des lignes de Georges Washington (GW_L) à partir des lignes elles-mêmes ou par progression de l'apprentissage des mots vers lignes.

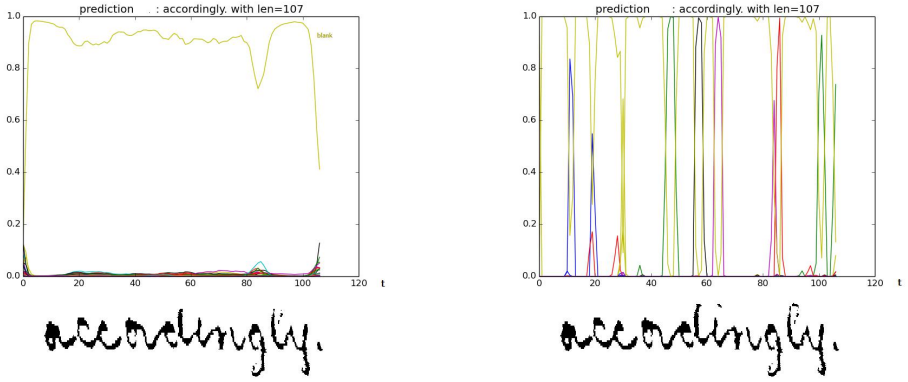
Table 4.2 – Résultats obtenus et comparaison avec l'état de l'art en reconnaissance d'écriture avec des réseaux utilisant le BLSTM-CTC sans lexique.

Système	Ressource App. et Test	Méthode	# Caract.	TRC	TRM	TRM_L
Graves et al. 2009	IAM_L	Locale	9	81,8 %		74,1 %
Morillot et al. 2013	RM_L	Locale	56	-		43,2 %
BLSTM-CTC	GW_M	Locale	4	69,62 %	38,65 %	
BLSTM-CTC		HOG	64	71,15 %	37,83 %	
BLSTM-CTC	GW_L	HOG	64	10,07%	-	
BLSTM-CTC	$(GW_M)_{ft}$ + GW_L	HOG	64	77,31 %		39,00 %

Nous avons réalisé ces premières expériences sans utiliser de lexique, pour faciliter le décodage, dans un premier temps. Dans notre contexte d'apprentissage par transfert de connaissances, nous souhaitons nous concentrer sur la reconnaissance exacte des séquences de caractères, car nous n'avons pas de connaissance a priori sur les informations contenues dans les images de la Comédie-Italienne. Ces premières expérimentations nous ont permis de constater que ce système était capable d'apprendre à reconnaître des séquences de caractères même s'il ne dépasse pas les TRC atteints sur d'autres ressources. Nous pouvons observer le comportement du modèle grâce aux figures 4.3a et 4.3b. À la première itération, le caractère *blank* est le plus actif à chaque instant (courbe plafonnant à 1 sur la figure 4.3a) et, au fil des itérations, l'activation de certains caractères par rapport à d'autres se dessine progressivement. Ce processus est présenté sur la figure 4.3b, où chaque courbe colorée correspond à un caractère de la séquence.

Nous pouvons comparer nos résultats sur GW_M avec l'étude de Lavrenko et al. 2004 qui utilise un système à base de HMM. Ce dernier atteint un TRM

1. IAM est constitué de 1 539 lettres en anglais avec 650 scripteurs différents. http://www.fki.inf.unibe.ch/databases/iam-handwriting-database



(a) "Itération 0" (b) "Itération 20"

FIGURE 4.3 – Figure : apprentissage de la séquence "accordingly" sur 20 itérations.

de 46,9% en excluant les mots hors-vocabulaire (de l'apprentissage) et avec 19 pages pour réaliser l'apprentissage. Nous obtenons un taux de mots correctement reconnus de 38,65% en nous plaçant dans les mêmes conditions. Cette différence de performance peut être attribuée à la faible quantité de données pour l'apprentissage du réseau de neurones alors qu'il nécessite plus de données comme cela a été prouvé dans (PANZNER et CIMIANO 2016) par rapport à un système de HMM.

La quantité d'images de lignes étant également très limitée sur Georges Washington, deux expériences ont été menées afin d'évaluer la capacité du système à reconnaître des lignes de mots (voir la partie inférieure du tableau 4.2). Pour la première expérience, le système a été entraîné directement sur les images de lignes durant 100 itérations. Dans la seconde expérience, les poids du réseau, entraînés sur les mots de Georges Washington, ont été utilisés pour initialiser et poursuivre l'apprentissage sur les images de lignes. Cette solution augmente de 67% le TRC en seulement 33 itérations. Cela a permis au système de s'attacher à reconnaître les caractères, dans un premier temps, puis d'identifier le caractère espace. Finalement, cette expérience nous montre qu'en reconnaissance de lignes, un apprentissage progressif des caractères est une bonne stratégie alternative.

Ces premières expériences nous ont ainsi permis de constater que notre réseau BLSTM-CTC, avec différentes méthodes d'extraction de caractéristiques, fonctionnait malgré la faible quantité de données et un risque non négligeable d'atteindre rapidement la zone de sur-apprentissage.

4.3.2.2 Données multilingues

Nous poursuivons les expériences en mode multilingue, dans cette section. Pour cela, nous utilisons uniquement l'extraction de caractéristiques par la méthode HOG avec la fenêtre glissante de 20 pixels et un pas de 2 pixels que nous utilisions précédemment. Afin d'améliorer la vitesse d'apprentissage du système, nous ordonnons les images de manière croissante suivant la longueur de leur transcription. Ainsi, à chaque itération, l'apprentissage se fait graduellement des caractères isolés vers les mots. De plus, en nous basant sur nos précédentes constatations, nous ne mélangeons pas les images de mots et les images de lignes. Dans un premier temps, l'apprentissage se fait sur les mots et les poids sont alors sauvegardés pour finalement être utilisés pour initialiser l'apprentissage réalisé avec les images de lignes.

Les données de GW_M sont régulièrement utilisées pour des tâches de détection automatique de mots (Rath et Manmatha 2007 ; Fischer et al. 2012 ; Frinken et al. 2012) plutôt que pour de la reconnaissance d'écriture de mots (Lavrenko et al. 2004). Cela est dû à la faible quantité de données qui n'est pas optimale pour réaliser l'apprentissage d'un réseau profond. Afin d'améliorer nos résultats précédents et de gérer cette quantité limitée de données, nous associons GW_M à RM_M qui est plus large.

La table 4.3 présente les premiers résultats obtenus. Pour ces expérimentations, nous utilisons une des répartitions des données fournies pour GW, c'est-à-dire avec 15 pages pour la phase d'apprentissage et de validation et 5 pages pour le test. Cela explique la baisse de performance que l'on observe sur la première ligne par rapport à la section précédente. La seconde ligne présente les résultats obtenus en utilisant conjointement les bases d'images de mots de RIMES et GW pour l'apprentissage, avec les tests effectués sur chacune des bases. On constate que la présence de RIMES a permis une légère amélioration sur le TRC ainsi que sur le TRM. Pour l'expérimentation suivante, l'utilisation des poids sauvegardés précédemment pour initialiser le modèle nous a permis d'observer qu'un modèle de langage est implicitement appris. En analysant les résultats en détail, nous avons constaté que le système fournit une séquence de caractères uniquement sur des mots très proches dans les deux langues : Français et Anglais. Le mélange des ressources est un point déterminant pour pouvoir transcrire nos documents de la Comédie-Italienne.

TABLE 4.3 – Résultats obtenus sur les ressources RIMES et Georges Washington . L'utilisation des poids d'un système comme initialisation d'un autre est indiquée par l'identifiant de l'expérience.

Apprentissage	Test	Id Exp	TRC	TRM	TRM_L
GW_M	GW_M	BC1	67,34 %	29,52 %	-
GW_M+	GW_M	BC2	69,91 %	32,01 %	-
RM_M	RM_M		79,11 %	42,98 %	-
BC2 +	GW_L	BC3	72,83 %	-	23,92 %
GW_L	RM_L		33,55 %	-	0,59 %

4.3.3 Transfert d'une langue à l'autre

Les nouvelles expériences présentées dans cette section utilisent plusieurs ensembles d'apprentissage pour pouvoir réaliser un apprentissage par transfert de connaissances. Il faut donc que les différentes parties composant le système soient capables de généraliser au mieux leur apprentissage. C'est pour cela que nous changeons de méthode d'extraction de caractéristiques en utilisant le RNC qui a la capacité de modéliser des données à différentes échelles et tailles.

4.3.3.1 Conditions d'apprentissage

Les premières expériences nous permettent d'optimiser l'architecture de notre système en l'évaluant avec un ensemble de données étiquetées. Avec les meilleurs paramètres obtenus, nous réalisons une autre expérience afin de sélectionner la meilleure combinaison de données et en vue de réaliser un apprentissage par transfert de connaissances sur les données Comédie-Italienne.

Dans l'apprentissage par transfert de connaissances, la capacité de généralisation du classificateur doit être maximisée donc nous continuons d'utiliser la méthode d'arrêt de l'apprentissage *early stopping* pour sélectionner le meilleur réseau : l'apprentissage continue jusqu'à ce que la fonction de vraisemblance, appelée *Negative Log-Likelihood* (NLL), calculée par le CTC ne diminue plus pendant 5 itérations sur l'ensemble de données de validation. Les poids du réseau sont sauvegardés pour chaque ensemble de données de validation uniquement si le NLL diminue. L'apprentissage se fait à travers toutes les parties du réseau : extraction de caractéristiques avec le RCN et système de reconnaissance de l'écriture manuscrite BLSTM-CTC.

Nos premières expériences sur GW ont démontré que l'entraînement s'effectuait plus efficacement sur les images au fur et à mesure de son déroulement. Par conséquent, toutes les images de l'ensemble d'apprentissage sont triées dans un ordre croissant selon leur longueur d'étiquette. De cette façon, l'apprentissage progresse des caractères isolés aux mots et finalement aux longues lignes. Premièrement, les expériences sont exclusivement exécutées sur les données de type mot. Ensuite, nous les étendons aux images de lignes. Cela permet d'augmenter la performance du BLSTM-CTC pour tous les ensembles de données.

4.3.3.2 Résultats

Optimisation de l'architecture du modèle Cette partie présente l'optimisation de l'architecture du système. Les paramètres de chaque partie de notre système (RCN et BLSTM-CTC) ont un impact sur la qualité des transcriptions ainsi que sur la phase de transfert. Ces expériences visent à évaluer les différents paramètres pouvant influencer notre système, c'est-à-dire la composition du RCN en termes de nombre de caractéristiques extraites et le nombre de cellules de type LSTM dans les couches masquées.

TABLE 4.4 – Résultats des différents modèles faisant varier la taille du RCN et le nombre de cellules composant les couches cachées du réseau, en utilisant le même ensemble d'apprentissage $\mathrm{RM}_M \cup \mathrm{GW}_M$.

Caractéristiques	Cellules LSTM	Test	TRC	TRM
32	50	GW_M	40,5	22,2
		RM_M	63,5	34,8
32	100	GW_M	43,9	22,2
		RM_M	67,6	39,9
128	50	GW_M	19,6	0
		RM_M	25,3	0,1
128	100	GW_M	48,5	25,5
		RM_M	70,1	41,7

Le tableau 4.4 présente les résultats obtenus par quatre systèmes entraînés avec les ensembles $\mathrm{RM}_M \cup \mathrm{GW}_M$ et testés sur les jeux de test RIMES $_M$ et GW_M. Les modèles se distinguent par leur nombre de caractéristiques (32 ou 128) ainsi que leur nombre de cellules dans les deux couches LSTM (50 ou 100).

Lorsque le nombre de caractéristiques extraites par le RCN est de 32, nous pouvons voir que l'augmentation du nombre de cellules LSTM conduit à un meilleur taux sur les caractères. Lorsque le nombre de caractéristiques est augmenté à 128, le taux de reconnaissance des caractères sur GW_M passe de 40,5% à 48,5%. La même observation peut être faite sur RIMES $_M$, où le taux de reconnaissance des caractères augmente de 6,6 %. Cependant, nous pouvons noter que l'augmentation du taux de reconnaissance des mots n'est pas aussi importante sur les deux ensembles de tests.

Les TRC obtenus sur GW_M avec l'extraction de caractéristiques automatique du RCN sont plus faibles de 20 à 39% qu'avec l'extraction par la méthode HOG. Cela peut s'expliquer facilement par la qualité des images puisque, pour ces expériences, nous utilisons les versions non pré-traitées des images de Georges Washington, c'est-à-dire en niveau de gris, avec le fond de l'image d'origine et le mot n'est pas bien centré. La différence sur le TRC de RIMES $_M$ est moins flagrante, car elle ne perd que 9%. Dans ce cas, cela s'explique par la propreté des images RIMES qui ont toutes un fond blanc.

Ces résultats nous permettent de conclure que la meilleure configuration est obtenue avec 128 caractéristiques extraites pour le RCN et avec les 100 cellules LSTM définies à l'origine par (Graves et Schmidhuber 2009). Ces paramètres sont conservés pour le reste de nos expériences.

Optimisation des données d'apprentissage L'architecture du système étant définie, nous devons définir les ensembles de données qui seront utilisés pour l'apprentissage de notre système. La quantité de données sur la Comédie-Italienne avec une vérité terrain est limitée et nous ne l'avons qu'au niveau mot et non au niveau ligne. Ainsi, parmi les trois ressources d'images sélectionnées, deux doivent être consacrées à l'apprentissage et la troisième doit jouer le rôle des données inconnues comme celles de la Comédie-Italienne. Nous avons observé précédemment qu'un modèle de langage s'apprenait implicitement. Pour pouvoir réaliser un apprentissage par transfert de connaissances, il est donc nécessaire que les combinaisons de ressources contiennent au moins une ressource en français et une ressource ancienne. La ressource RIMES doit faire partie des données d'apprentissage, car il s'agit du seul ensemble de données français dont nous disposons.

Parmi les études réalisées sur RIMES, les auteurs de (Pham et al. 2014) utilisent un réseau profond multidimensionnel ainsi qu'un RCN. Les auteurs font varier le nombre de cellules LSTM de 30 à 200. Elles obtiennent un TRC de 84,9% avec 50 cellules et un TRC de 84,2% avec 100 cellules. En comparant cela, avec nos résultats présentés dans le tableau 4.5, nous constatons que nos différents apprentissages ont un TRC de 13% de moins. Cela peut s'expliquer facilement par la taille du

TABLE 4.5 – Résultats obtenus avec le *RNC128* et 100 cellules LSTM dans les couches cachées pour différentes combinaisons de ressources d'images de mots.

Id	Apprentissage	Test	TRC	TRM
Exp_1	$\text{RM}_M \cup \text{GW}_M$	GW_M	48.5	25.5
		RM_M	70.1	41.7
		ESP_M	9.0	0.3
Exp_2	$\text{RM}_M \cup \text{ESP}_M$	GW_M	6.3	0.3
		RM_M	71.1	42.0
		ESP_M	91.1	75.9

réseau mis en place dans notre étude qui se veut plus simpliste et plus généraliste. De plus, le changement de ressources entre GW et ESP ne permet pas d'observer une amélioration sur les résultats de RIMES.Les TRC sont élevés sur les ensembles de tests de RIMES et ESP en utilisant ces mêmes ressources pour l'apprentissage. Cela est dû à la quantité de données, car les ensembles cumulés fournissent plus de 100k images de mots. Nous sélectionnons ce couple de ressources pour réaliser la dernière série d'expériences sur les données de la Comédie-Italienne. Ces premières expériences sur l'apprentissage par transfert de connaissances au niveau des mots montrent qu'il s'agit d'une tâche difficile.

Apprentissage par transfert de connaissances sur la Comédie-Italienne. Les expériences précédentes nous ont permis de configurer notre système afin qu'il soit le plus performant sur un apprentissage classique. Ces dernières expériences mettent en œuvre un processus d'apprentissage par transfert de connaissances à deux échelles sur les images de lignes, et tout particulièrement celles de la Comédie-Italienne. Nous souhaitons également évaluer l'impact que peut avoir l'ajout ou non de données cibles lors de l'apprentissage afin de spécialiser le réseau généraliste. Le tableau 4.6 présente les résultats de trois expériences différentes réalisées sur l'ensemble de test de la Comédie-Italienne RCI-RC. Chacune des expériences est initialisée avec les poids sauvegardés d'une expérience précédente (indiquée par un identifiant dans la colonne Apprentissage).

Notre cas utilisateur correspond à la première ligne. Les résultats présentés sur les lignes 2 et 3 ont été obtenus en utilisant le sous-ensemble de RCI-RC dédié à l'apprentissage. Dans un premier temps, nous utilisons les poids sauvegardés de la dernière expérience Exp_2 sur les mots, présentée dans le tableau 4.5, et nous ajoutons la ressource RM_L tout en conservant ESP_M. Au cours des expériences, nous avons découvert que l'initialisation du réseau avec des poids sauvegardés n'est utile que si les ressources utilisées pour configurer les poids sont toujours

TABLE 4.6 – Résultats obtenus avec le *RNC128* et 100 cellules LSTM dans les couches cachées utilisant les poids sauvegardés d'expérimentations précédentes et évalués sur RCI-RC.

Id	Apprentissage	Test	TRC
Exp_3	$(Exp_2) \cup$ $\text{RM}_L \cup \text{ESP}_M$	RCI-RC_L	10.6
Exp_4	$(Exp_2) \cup \text{RCI-RC}_L$	RCI-RC_L	25.5
Exp_5	$(Exp_3) \cup \text{RCI-RC}_L$	RCI-RC_L	28.7

présentes dans l'apprentissage. En effet, le système oublie. Dans un second temps, les poids sauvegardés lors de cette expérience sont utilisés pour initialiser le réseau pour les expériences sur RCI-RC_L. Cette spécialisation du réseau sur les données de la Comédie-Italienne lors de l'apprentissage permet d'augmenter le taux de reconnaissance des caractères de 15%. Enfin, afin d'observer l'apprentissage au préalable du caractère " espace " lors de l'apprentissage sur les lignes de RIMES, et l'impact potentiel de la quantité d'images utilisées, les poids sauvegardés de Exp_3 (incluant cet apprentissage sur les lignes) sont choisis pour initialiser le réseau et tester sur RCI-RC_L. Nous constatons que le TRC est augmenté de 3%.

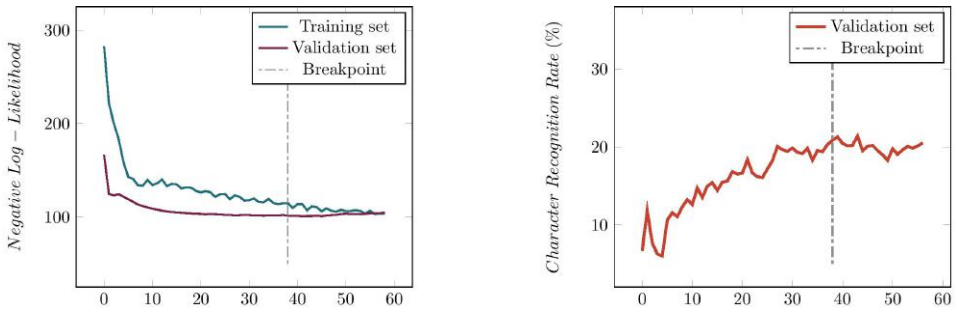


FIGURE 4.4 – Fine-tuning de RCI-RC_L avec Exp_2 : évolution de Log-Vraissemblance par itération

Les figures 4.4 et 4.5 montrent les courbes obtenues lors des différents apprentissages sur RCI-RC_L, à partir des images de mots uniquement contre les images de lignes RIMES et celles des mots de Les Esposalles.En plus d'améliorer le taux de reconnaissance des caractères, nous notons que l'apprentissage est plus rapide (38 itérations contre 27) en utilisant l'apprentissage avancé avec RM_L et non uniquement RM_M. De plus, la valeur initiale de NLL à l'itération 0 est deux fois plus faible avec Exp_3 et elle atteint rapidement un niveau de stabilité lorsque le caractère espace a été appris auparavant. Avec l'initialisation par Exp_2, le TRC

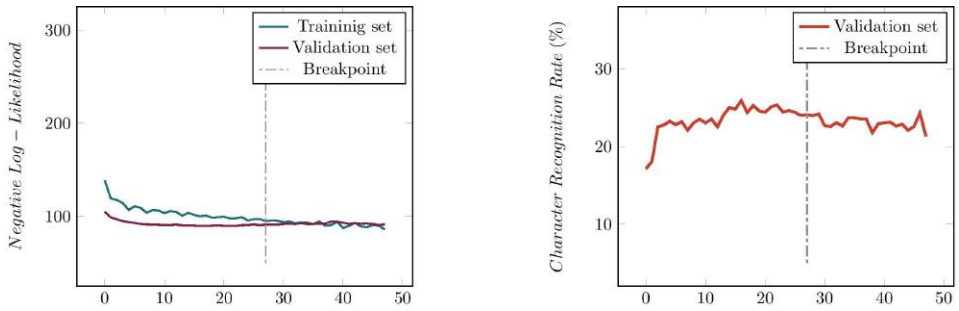


FIGURE 4.5 – Fine-tuning de RCI-RC$_L$ avec Exp_3 : évolution de Log-Vraissemblance par itération

sur l'ensemble de validation augmente progressivement. Nous pouvons constater une chute brutale au début de la courbe d'apprentissage : il semble que le système oublie une partie de ce qu'il a appris pour pouvoir se spécialiser sur RCI-RC$_L$. Avec Exp_3, la courbe de TRC semble poursuivre l'apprentissage dans la continuité de la précédente expérience. Il est également intéressant de noter que le point d'arrêt sur la NLL de validation est juste après la meilleure valeur atteinte pour le TRC. En outre, nous pouvons voir que, avec une seule itération sur un petit ensemble de données, le TRC de l'expérience Exp_5 est plus élevé que lors de l'expérience Exp_4.

TABLE 4.7 – Exemple de résultats obtenus sur une ligne de la Comédie-Italienne avec l'expérience Exp_5.

Image	
Hypothèse	La deies oene, e eer ae

La table 4.7 montre un exemple de résultat obtenu pour le titre "La Bohemienne, La fille mal gardee". Le TRC est de 40 % pour une longueur totale de 34 caractères, 12 insertions, 1 suppression et 8 substitutions. Nous pouvons constater que les espaces et la ponctuation ont correctement été reconnus, et que les voyelles sont les caractères les plus prédits.

4.4 Conclusion

Nous avons effectué nos premières expériences pour l'apprentissage par transfert de connaissances en reconnaissance d'écriture manuscrite sur des données historiques de la Comédie-Italienne. Tout d'abord, nous avons optimisé les paramètres du

système afin d'obtenir le système le plus simple et le plus performant qu'il soit. Ensuite, nous avons défini la meilleure combinaison des ressources pour réaliser l'apprentissage par transfert de connaissances. De plus, il faut faire attention à l'équilibre de la représentation des mots dans un jeu de données, car cela a un impact important sur le système. La dernière expérience avec les données de la Comédie-Italienne montre que, même si les taux de reconnaissance sont faibles, nous notons une progression.

Nos expériences nous permettent de conclure que les modèles de reconnaissance d'écriture se spécialisent rapidement dans l'apprentissage des données. Nous avons remarqué que l'ajout d'une ou de plusieurs ressources permettait d'améliorer la reconnaissance des caractères. Il semble nécessaire d'ajouter une petite quantité de données du domaine cible dans l'apprentissage pour atteindre un taux de reconnaissance suffisant. Notre objectif reste de pouvoir effectuer un apprentissage par transfert de données sans connaissance préalable coûteuse. Ces résultats préliminaires sur l'apprentissage par transfert de connaissances ont été publiés dans la conférence ICPRAM (Granet et al. 2018b).

Au vu des résultats, nous pouvons remettre en doute l'architecture élaborée pour réaliser la reconnaissance d'écriture. Le modèle de langage appris implicitement semble venir de l'utilisation des cellules LSTM qui conservent les informations en mémoire sur du long terme. L'apprentissage d'un réseau global tel que nous l'avons défini ne paraît pas fonctionner. Forts de cette constatation, nous avons décidé de séparer en deux modèles bien distincts le réseau avec une partie dédiée à l'image et l'autre au langage. Cette orientation nous permet d'envisager l'apprentissage du réseau différemment avec de nouvelles ressources de nature différentes. Le chapitre suivant présente ce nouveau modèle en détail dans lequel une section est dédiée à un des composants du modèle, avant de présenter les résultats sur le modèle complet.

5

Système encodeur-décodeur pour l'apprentissage par transfert de connaissances

Sommaire

5.1 Introduction

Dans le chapitre précédent, nous avons constaté que l'apprentissage par transfert de connaissances était difficile à opérer avec un système classique BLSTM-CTC, comme l'ont montré les expériences. Un système qui fournit de bons résultats en reconnaissance de caractères et de mots, sur une ressource pour laquelle le réseau a été entraîné, n'induit pas qu'il sera performant pour une autre ressource sans *fine-tuning*. Un changement de stratégie est nécessaire pour pouvoir obtenir des résultats. Il existe différentes solutions envisageables pour apporter les améliorations nécessaires pour de l'apprentissage par transfert de connaissances.

Une solution consiste à augmenter la taille du réseau pour se rapprocher des réseaux multidimensionnels de l'état-de-l'art qui ont fait leurs preuves comme (Bluche et al. 2017b). Cependant, les résultats obtenus précédemment ont montré que des réseaux simples par leur architecture sont capables d'obtenir de bons résultats sur des données provenant de la même ressource que celle utilisée en apprentissage. Ils montrent également des signes d'apprentissage du langage bien qu'un arrêt précoce de l'apprentissage ait été mis en place. C'est pour cette raison que nous avons écarté la solution visant à complexifier le système. Nous avons préféré nous tourner vers une autre architecture qui a fait ses preuves dans d'autres domaines : le modèle encodeur-décodeur.

Les modèles *sequence-to-sequence* sont très présents dans le traitement du langage et plus particulièrement dans les domaines de la traduction automatique (TA) (Cho et al. 2014a), des questions-réponses (Q&A), et des agents conversationnels (Qiu et al. 2017), (Chan et al. 2016). Depuis 2013, le modèle type encodeur-décodeur basé sur un réseau neuronal récurrent, proposé par Cho et al. 2014b domine l'ensemble de ces domaines. Le premier composant encode la source de taille variable en un vecteur de taille fixe appelé *Thought vector*. Le second composant décode la séquence dans la cible qui peut avoir une longueur ou un ordre de mots différent. Cette séquence peut être dans une langue différente pour la TA (Bahdanau et al. 2014), dans une forme différente en reconnaissance de la parole par exemple (parole vers écrit) ou dans la même langue pour Q&A. De même, Vinyals et al. 2015 ont proposé un générateur neuronal de description d'images constitué de deux sous-réseaux : un réseau neuronal convolutif pré-entraîné encodant une image dans un vecteur de taille fixe, en utilisant la dernière couche

cachée de GoogleNet, suivi par un modèle LSTM générant la description correspondante. Les réseaux sont construits à partir de réseaux récurrents dont l'apprentissage se fait conjointement pour maximiser la probabilité conditionnelle entre la source et la cible. De plus, le vecteur faisant le lien entre les deux réseaux représente un espace latent.

La contribution majeure que nous proposons est une démarche exploratoire autour de l'architecture encodeur-décodeur, comme modèle de reconnaissance d'écriture manuscrite à partir de deux réseaux complémentaires. Nous souhaitons aller plus loin que l'état-de-l'art en permettant un apprentissage disjoint des deux composants. Cela implique de définir un espace commun non-latent pour représenter la sortie de l'encodeur qui sera également l'entrée du décodeur. Autrement dit, nous souhaitons définir un espace dans lequel chaque position du vecteur correspond à une information bien définie. En comparaison avec un modèle encodeur-décodeur classique, les informations d'une séquence sont compressées, dans un espace réduit par rapport à la séquence d'entrée, pour ensuite être décompressées par le décodeur. C'est cet espace réduit que l'on appelle non-latent.

Ce chapitre présente, dans un premier temps, l'idée globale du modèle mis en place en définissant chaque composant, leur rôle ainsi que leur représentation. Définir ainsi des composants indépendants, mais complémentaires permet de les évaluer séparément. Une deuxième partie sera dédiée à l'élaboration de l'encodeur avec les résultats obtenus lors des différentes expérimentations et une troisième partie se focalisera sur le décodeur ainsi que sur les résultats obtenus lors des expérimentations. Finalement, la dernière partie présentera la mise en relation des deux réseaux.

5.2 Définition du modèle encodeur-décodeur

Nous avons choisi de mettre en place le modèle encodeur-décodeur proposé en figure 5.1, composé de deux réseaux indépendants, mais complémentaires partageant un espace commun. Le premier réseau est l'encodeur d'image qui, comme son nom l'indique, permet de projeter l'image qu'il prend en entrée dans un espace non-latent que nous avons défini. Puis, à partir de ce vecteur pivot, le décodeur génère une séquence de caractères correspondant à la transcription de l'image. Pour donner un sens au *thought vector*, nous avons choisi d'y représenter des n-grammes de caractères. (Huang et al. 2013a ; Bengio et Heigold 2014) ont utilisé cette représentation dans des contextes différents mais son utilisation semble pertinente et justifiée dans notre cas d'étude. D'autres approches récentes en *word spotting* et reconnaissance d'écriture manuscrite ont également utilisé une projection dans un espace commun

entre les étiquettes et les caractéristiques des images comme (Almazán et al. 2014b).

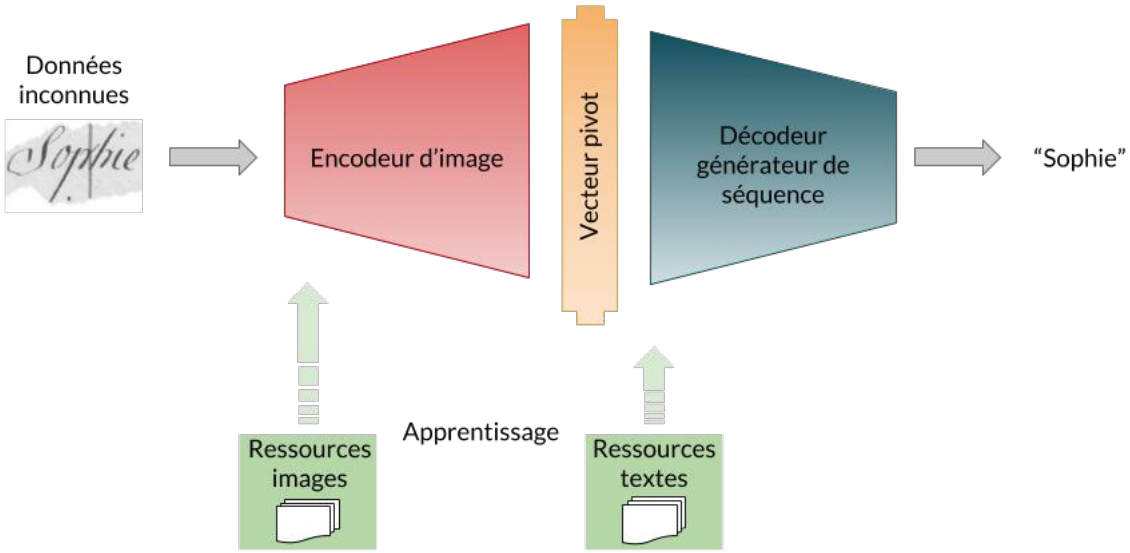


Figure 5.1 – Modèle encodeur-décodeur réalisé pour expérimenter l'apprentissage par transfert de connaissances avec un vecteur de n-grammes de caractères.

L'originalité de notre approche réside dans le fait d'utiliser ce vecteur de n-grammes comme pivot du système et de supprimer la notion de temporalité entre les caractères d'un mot. Ce vecteur permet d'encoder les informations dans un espace non-latent qui est transférable tant que les données d'apprentissage et les données de cible de transfert partagent le même alphabet. Une telle approche favorise un apprentissage indépendant des deux parties du système.

Le transfert de connaissance est opéré à plusieurs niveaux : ressources et paramètres. Rappelons que, par définition, l'apprentissage transductif par transfert de connaissance se fait pour une tâche donnée, avec des données sources et cibles différentes, mais de mêmes domaines et des données cibles inconnues. Comme pour le premier modèle présenté dans le chapitre précédent, c'est ce que nous mettons en place ici avec les ressources d'images annotées disponibles, auxquelles nous ajoutons de nouvelles données de type linguistique pour le décodeur. Bien que nous réalisions un apprentissage indépendant des deux composants, il est également possible d'initialiser les poids des modèles pour entraîner le système complet à travers les deux composants.

Définition du vecteur pivot L'élément pivot de notre encodeur-décodeur est un vecteur de n-grammes de caractères. L'utilisation de cette représentation fait référence aux n-grammes de caractères utilisés comme modèles de langages qui ont largement fait leurs preuves par le passé. Une étape importante est la définition de ces n-grammes de caractères. En considérant un langage contenant 75 symboles dans son alphabet, le vecteur pivot aurait une taille de $\sum_{i=1}^{N} 75^i$ où N est la longueur maximum des n-grammes. Par exemple, avec une longueur maximum de 3 pour la sélection des n-grammes de caractères, le vecteur aurait une taille de $75 + 75^2 + 75^3$, soit $4,27.10^5$ n-grammes de caractères. Cette dimension est beaucoup trop grande pour être envisageable. Il faut donc opérer une sélection plutôt que de considérer l'ensemble des n-grammes possibles. Par exemple, les auteurs de (Bengio et Heigold 2014) ont sélectionné les 50 k n-grammes les plus fréquents de longueur maximum 3, pour construire leur vecteur.

Pour pouvoir sélectionner les n-grammes de caractères qui composent le vecteur pivot, nous nous sommes basés sur les ressources d'apprentissage présentées dans le chapitre 3. Nous avons conservé la limite d'une longueur de 3-grammes au maximum comme cela a été réalisé dans les études précédentes, mais également, car des études autour des modèles de langage et de la représentation des mots ont montré que les meilleurs résultats étaient obtenus avec cette longueur (Vania et Lopez 2017 ; Huang et al. 2013a). En reconnaissance d'écriture manuscrite, (Bluche et al. 2017a) ont également travaillé sur des bigrammes, mais non adjacents pour reconnaître des mots et ont obtenu des résultats similaires aux méthodes classiques. Pour chaque ressource, nous décomposons tous les mots en n-grammes de caractères en incluant les symboles de début de mot *[* et fin de mot *]*. Par exemple, la décomposition du mot *Sophie* de la figure 5.1 est {S,o,p,h,i,e,[S, So, op, ph, hi, ie, e], [So, Sop, oph, phi, hie, ie]} soit un total de 19 n-grammes. D'une manière générale, nous avons ainsi $3n + 1$ n-grammes de longueur maximale 3 pour un mot de n caractères.

Nous avons déjà mentionné le caractère bruité de la ressource GCI. Les prétraitements réalisés au moment de l'extraction du vocabulaire ont laissé du bruit. Pour remédier à cela, nous rejetons tous les n-grammes estimés avec une fréquence strictement inférieure à 2 occurrences dans l'ensemble du corpus. La figure 5.2 montre la répartition des différents n-grammes que nous avons pu estimés suivant leur nombre d'occurrences dans la ressource GCI. La quantité d'unigrammes est la seule quantité à être fixe. Nous constatons qu'à chaque fois que l'on augmente la borne minimale de la fréquence des n-grammes, une perte significative d'éléments est observée. Nous avons démontré précédemment qu'avec une longueur maximum de 3 pour un vocabulaire de 75 caractères, il existe au total plus de $4,27.10^5$ n-grammes. En ne sélectionnant que les n-grammes ayant une fréquence minimale

de 2 dans notre corpus GCI, nous ne conservons que 2,5% de l'ensemble des $4,27.10^5$ n-grammes, tandis qu'avec une fréquence minimale de 10, la quantité de n-grammes sélectionnés correspond à 1,67% de l'ensemble des n-grammes. Il est évident que c'est trop peu pour pouvoir représenter un large vocabulaire et dans différentes langues. C'est pour cette raison que nous avons opté pour sélectionner les n-grammes de caractères avec une fréquence minimale de 2. Cela permet d'éliminer un minimum de bruit restant tout en maximisant la taille du vecteur pivot en se basant sur les ressources.

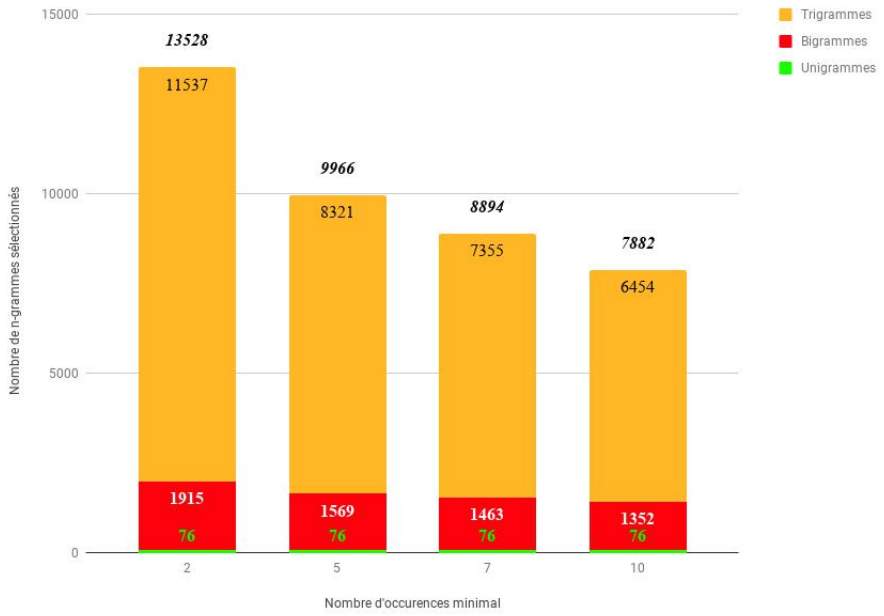


FIGURE 5.2 – Répartition des différents n-grammes en fonction du nombre d'occurrences sélectionnées.

L'estimation des n-grammes est opérée sur chacun des quatre corpus GCI, RIMES, ESP et GW indépendamment les uns des autres en prenant tous les n-grammes de caractères sans distinction (excepté pour GCI). Puis, nous conservons les n-grammes présents au moins dans 2 ressources différentes pour venir constituer le vecteur final. Cette contrainte nous permet de limiter le nombre de n-grammes de caractères tout en éliminant des n-grammes non-fréquents. Finalement, nous obtenons 12 179 n-grammes mélangeant majuscules, minuscules, caractères de ponctuations et chiffres. De plus, pour l'ensemble des ressources, nous avons remplacé les caractères accentués par leur forme simple, par exemple [é,è, ê,ë] par le caractère "e". Nous avons également remplacé les formes spéciales, comme la forme longue du "s", typique du XVIII^e siècle, en sa forme courte afin d'harmoniser les

représentations multilingues et multi-époques. Nous ajoutons enfin, à l'ensemble des n-grammes, un dernier élément appelé *joker* pour remplacer les n-grammes non-sélectionnés.

Si nous comparons la taille du vecteur créé (12 170) avec celui utilisé dans l'état de l'art (50 000), cela semble petit et la quantité d'information représentée limitée. Cependant, avec les expériences, nous nous rendrons compte que la difficulté imposée par cette taille est déjà assez grande.

Après la sélection de cet ensemble de n-grammes de caractères, il faut également choisir leur représentation au sein du vecteur par rapport aux mots. Une possibilité est d'utiliser une information binaire notifiant la présence ou non d'un n-gramme de caractères dans le mot, mais cela fait perdre une information importante pour la reconstruction du mot qui est la taille de ce dernier. Une autre possibilité, qui est également celle utilisée par les auteurs (Huang et al. 2013b), est d'indiquer la quantité de chaque n-gramme dans le mot sans normalisation. Nous faisons le choix de construire notre vecteur par normalisation de la fréquence de chaque n-gramme présent dans le mot, malgré la difficulté que cela peut représenter. Cela permet de conserver une information sur la taille du mot et de compenser la séquentialité supprimée.

Nous venons de décrire le vecteur de n-grammes de caractères qui est le pivot et l'enjeu de notre modèle encodeur-décodeur. Dans la prochaine section, nous allons définir en détail le modèle optique.

5.3 Modèle optique comme encodeur

Le réseau mis en place comme modèle optique diffère très peu de ce qui est réalisé en tant que modèle de reconnaissance d'écriture. L'encodeur prend en entrée une image, dans laquelle il faut extraire des caractéristiques. La différence majeure est la sortie qui fournit un vecteur indiquant les n-grammes de caractères identifiés dans l'image à la place d'une séquence de caractères. La figure 5.3 offre une vue détaillée de l'architecture finale réalisée : elle est expliquée à la section 5.3.1, et les résultats obtenus sont donnés à la section 5.3.2.

5.3.1 Architecture du modèle

Pour obtenir une architecture satisfaisante, nous avons expérimenté différentes stratégies afin d'obtenir les meilleurs résultats. Certains éléments sont restés fixés tout au long du processus, comme la structure de base du réseau à convolution pour

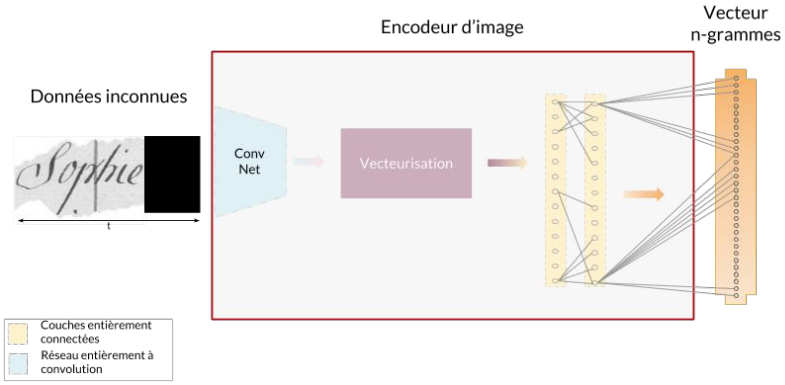


FIGURE 5.3 – Fonctionnement et conception du modèle optique.

extraire les caractéristiques ainsi que les couches entièrement connectées précédant le vecteur.

Extracteurs de caractéristiques Le premier élément constituant ce modèle est un réseau de neurones entièrement à convolution (RNC) qui extrait et construit des caractéristiques à partir des images de mots passées en entrée. Nous avons pris le parti de ne pas utiliser de réseau pré-entraîné existant. En effet, les réseaux existants, dédiés à cette tâche ou utilisables, ont été entraînés sur des images d'objets et de paysage en couleurs naturelles et non du texte. Or, nous travaillons sur des images en niveaux de gris. De plus, en cumulant l'ensemble des ressources d'images sélectionnées, nous avons plus de 100k images. Ainsi, nous avons préféré entraîner notre propre réseau pour extraire les caractéristiques des images. L'architecture du RNC construit est détaillée dans la table 5.1. La taille de l'image d'entrée est normalisée à 100 pixels de hauteur tout en conservant la proportion originale de l'image entre la longueur et la hauteur. Ainsi, la taille du vecteur de caractéristiques extrait varie en fonction de la longueur de l'image t. Ce réseau est composé de 7 convolutions avec chacune une taille de 3 par 3 pour le noyau. De plus, l'option *same-padding*[1] est utilisée, ce qui permet de conserver la taille d'entrée en sortie de la couche pour la longueur t et la hauteur. Certaines couches de convolution sont liées par des couches dites de *Max-Pooling* pour réduire la taille de l'objet en prenant la valeur maximum dans le filtre défini. Ceci conduit à une représentation dépendant uniquement de la longueur de l'image et du nombre de filtres utilisés pour extraire les caractéristiques. Chaque couche de convolution utilise la fonction d'activation *Rectify Linear Unit* (ReLU) (NAIR et HINTON 2010) définie par

1. Cette option, proposée par Lasagne, ajoute des 0 autour de la matrice d'entrée en fonction de la taille du noyau choisi.

$ReLU(x) = max(0, x)$ où x est la somme des entrées pondérées de chaque neurone, qui a l'avantage d'accélérer l'apprentissage.

TABLE 5.1 – Architecture du réseau de neurones entièrement à convolution. Les noyaux des couches utilisés sont au format (noyau)@ nb filtres. Les sorties des couches sont notées au format $h \times t$@ nb filtres

Nom	**Type de couche**	**Taille du fitre**	**Format en sortie**	**Fonction d' Activation**
Image	Image d'entrée	///	100×1×t	///
Conv 1	Convolution	(3×3)@8	$(100 \times t)$@8	ReLU
Conv 2	Convolution	(3×3)@16	$(100 \times t)$@16	ReLU
Conv 3	Convolution	(3×3)@32		ReLU
Maxpool 3	+ MaxPooling	(2×2)	$(50 \times \frac{t}{2})$@32	
Conv 4	Convolution	(3×3)@64		ReLU
Maxpool 4	+ MaxPooling	(2×2)	$(25 \times \frac{t}{4})$@64	
Conv 5	Convolution	(3×3)@128		ReLU
Maxpool 5	+ MaxPooling	(5×5)	$(5 \times \frac{t}{20})$@128	
Conv 6	Convolution	(3×3)@256	$(5 \times \frac{t}{20})$@256	ReLU
Conv 7	Convolution	(3×3)@512		ReLU
MaxPool 7	+ MaxPooling	(5×1)	$(1 \times \frac{t}{20})$@512	

Modélisation des caractéristiques Le vecteur de n-grammes de caractères à construire doit pouvoir fournir l'information de début et fin de mot, ainsi que la décomposition en n-grammes de caractères contenus dans l'image. Le tout doit être concentré dans un vecteur de taille fixe. Comme nous l'avons également mentionné dans la figure 5.3, ce sont des couches entièrement connectées qui précédent directement la formation du vecteur final. Ces couches requièrent une entrée à une dimension de taille fixe. Or, nous venons de montrer que la sortie du RNC est une matrice de taille $(1 \times \frac{t}{20})$@512. Il faut donc réaliser une opération de vectorisation pour que les caractéristiques extraites soient dans la bonne dimension pour les couches suivantes. En plus de devoir aplatir les informations, nous avons besoin de gérer, voire localiser la position des différents n-grammes composant l'image pour définir leur fréquence.

La première solution mise en place est une structure pyramidale réalisée à partir d'une couche de neurones appelée *Spatial Pyramid Pooling DNN Layer* initialement proposée par (HE et al. 2014). Cette couche transforme un objet en deux dimensions de n'importe quelle taille en un vecteur de taille fixe. Cela offre l'avantage de pouvoir

créer une liaison entre une couche de convolution (associée à une couche de *Max-Pooling*) et une couche dense comme dans notre modèle. Une liste de 3 niveaux est définie à l'initialisation de la couche spatiale pyramidale, ainsi que la fonction pour réaliser le sous-échantillonnage sur l'ensemble des niveaux. Pour chaque niveau défini l, les caractéristiques en entrée qui sont de dimension $(h \times t)$@512 sont segmentées en $n_l \times n_l$ afin d'appliquer la fonction d'échantillonnage. Cela produit une représentation fixe des caractéristiques directement proportionnelle aux nombres de caractéristiques extraites sur la dernière couche de convolution Conv 7 (sans la couche MaxPool 7 qui la succède normalement, car il est impossible de diviser la sortie de $(1 \times \frac{t}{20})$). Pour chaque niveau, la sortie obtenue est de dimension $(n_l \times n_l)$@512. Nous avons choisi les niveaux 1,2,4 pour réaliser la pyramide de caractéristiques, qui sont les valeurs proposées par défaut, en faisant intervenir la fonction calculant le maximum dans les zones. Nous obtenons en sortie un vecteur de taille $21@512 = (4 \times 4 + 2 \times 2 + 1)$ @512. Cette solution présente un avantage permettant la transition vers une couche entièrement connectée. Cependant, il existe aussi des inconvénients. Nous travaillons sur des images de grandes tailles (en longueur) donc il y a une perte d'information lors du calcul du sous-échantillonnage. Le second problème notable est la segmentation en zones. Comme le montre la figure 5.4, chaque couche de caractéristiques est découpée selon la longueur totale de la sortie du RCN indépendamment de la position de l'objet. Cela peut provoquer une perte d'information potentielle puisque, dans certains segments, il peut n'y avoir aucune information sur le mot tandis que, dans d'autres, il peut y en avoir trop et donc les informations se perdront dans la masse.

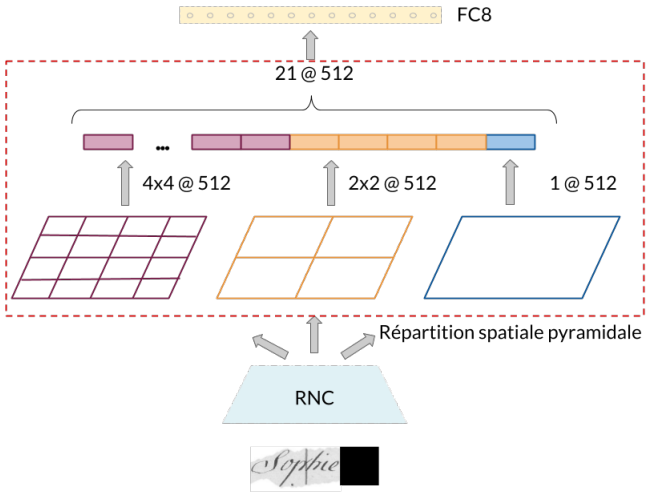


FIGURE 5.4 – Représentation de la couche de sous-échantillonnage pyramidale spatiale comme liaison entre le RNC et FC8 de notre modèle optique.

La seconde solution utilise des masques définissant des zones dans les images pour simuler un modèle d'attention artificiel. C'est cette approche qui est présentée sur la partie gauche de la figure 5.5. Pour chaque image passée en entrée du réseau, nous calculons un ensemble de masques afin de cibler localement les caractéristiques extraites. Le fait de se focaliser sur certaines zones est similaire à un modèle d'attention. Pour être efficace, le vecteur de n-grammes de caractères a besoin d'identifier les n-grammes de début et de fin de mot, mais également une information globale sur le mot. Pour cela, nous définissons trois masques différents :

— un masque fournissant la position du mot dans l'image ;
— deux masques fournissant les positions de début et fin de mot dans l'image ;
— trois masques fournissant les positions de début, milieu et fin de mot dans l'image.

Chaque zone délimitée par un masque est calculée par rapport à la longueur du mot dans l'image et non par rapport aux nombres de caractères le composant. Ainsi, nous restons dans le cadre de données inconnues et nous travaillons sans apriori sur les mots. Cette stratégie est élaborée et fonctionne pour des mots isolés. Finalement, nous obtenons six masques par image permettant de faciliter l'ordre des n-grammes de caractères. La longueur de l'image change à mesure que les convolutions et les sous-échantillonnages sont opérés. Elle est réduite à $\frac{t}{20}$ en sortie du RCN. Cette information est intégrée dans le calcul des masques puisqu'ils sont appliqués après l'extraction de caractéristiques. Chacun de ces masques est ensuite fusionné à l'aide d'une opération de multiplication avec les caractéristiques extraites sur l'image.

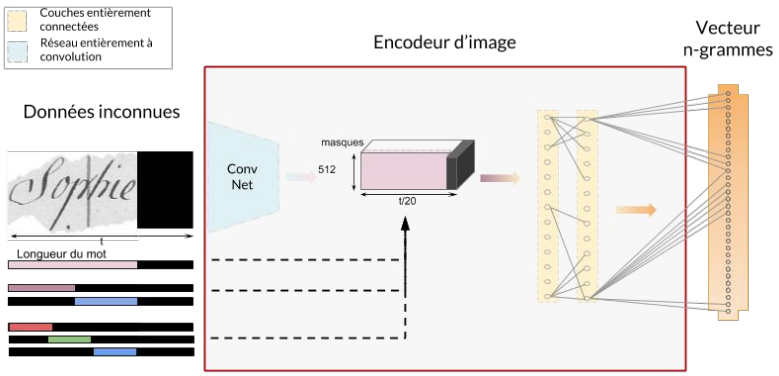


FIGURE 5.5 – Fonctionnement et conception du modèle optique en utilisant les 6 masques construits à partir de l'image d'entrée.

La comparaison des deux solutions proposées dans cette section pour segmenter les caractéristiques obtenues est présentée dans la section 5.3.2.

Estimation du vecteur de n-grammes de caractères Le modèle optique se termine par deux couches denses entièrement connectées précédant la dernière couche calculant le vecteur de n-grammes de caractères. La première est composée de 1 024 neurones, tandis que la seconde a une taille qui varie jusqu'à 2 048 neurones. La dernière couche construit le vecteur fournissant la fréquence de chaque n-grammes de caractères pour un mot. Nous avons remarqué que la fréquence ne dépasserait pas la borne de 0,25. En effet, le mot le plus court que nous pouvons avoir est composé d'un unique caractère ce qui se décompose en 4 n-grammes de caractères : {[a,a,a],[a]}. Cela implique une très faible différence entre les n-grammes présents et non présents. C'est pour cette raison que nous avons mis en place plusieurs stratégies, pour l'activation des neurones de la dernière couche ainsi que pour calculer la valeur cible, qui vont définir la bonne construction du vecteur de n-grammes. Il est important pour le décodeur que l'identification des n-grammes soit la plus optimum.

SIG Une dernière couche entièrement connectée avec une fonction d'activation Sigmoïde définie comme

$$Sigmoide(x) = \frac{1}{1 + e^{-x}}$$

pour obtenir l'information réelle de chaque n-gramme indépendamment des autres. L'information binaire est utilisée pour cette fonction, les n-grammes appartenant au mot sont à 1 et les autres 0.

SIG OFF Une dernière couche entièrement connectée avec une fonction d'activation sigmoïde utilisant un décalage de 0,5 pour les n-grammes de caractères présents dans le mot. Pour cela, au moment du calcul du vecteur cible durant l'apprentissage, à chaque probabilité non nulle, nous avons ajouté cette valeur afin d'obtenir une différence un peu plus significative pour le réseau.

THR Une couche entièrement connectée, avec une fonction d'activation tangente hyperbolique définie comme

$$\tanh(x) = 2 \times Sigmoïde(2x) - 1$$

, permet de fournir des valeurs tendant vers -1 pour les n-grammes non présents (à la place de 0 avec SIG) et tendant vers 1 pour les n-grammes présents dans l'image de mot. Cette fonction présente l'avantage d'accroître la différence entre les n-grammes. Pour pouvoir aligner les résultats de la sortie avec la vérité terrain, cette couche est suivie par une couche appliquant une fonction d'activation ReLU sans apprentissage des paramètres et fournissant des valeurs de sortie entre 0 et 1.

Fonctions objectifs Nous avons expérimenté plusieurs architectures pour le modèle, il est donc important de trouver la fonction objectif la plus appropriée pour chaque modèle. Nous évaluons le modèle à l'aide d'une fonction de perte qui tend à minimiser la différence entre la prédiction du modèle et la valeur réelle de chaque sortie. Dans ce but, nous avons défini deux fonctions de perte différentes.

Erreur quadratique moyenne est définie par la fonction suivante :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(T_i - P_i)^2$$

Le calcul de l'erreur moyenne est la méthode usuelle lorsqu'il s'agit de réaliser une régression linéaire. Pour chaque observation, la différence entre la valeur observée et la valeur prédite par le système est faite, avant de réaliser la moyenne des carrés de l'ensemble des différences.

Entropie est définie pour un processus de Bernoulli (binaire) par la fonction suivante (*binary entropy*) :

$$H(p,q) = -\sum_{i=1}^{n} P_i \log T_i$$

Elle compare directement chaque sortie obtenue avec la vérité terrain indépendamment des autres sorties. Elle est fortement conseillée dans le cadre de problème utilisant une sigmoïde. Cependant, nos valeurs cibles ne sont pas de simples valeurs binaires (pour l'ensemble des expériences), mais des fréquences.

Kullback-Leibler est définie par la fonction suivante :

$$D(P \parallel T) = \sum_{i=1}^{n} P_i \log \frac{P_i}{T_i}$$

Contrairement aux deux autres fonctions, cette dernière calcule la divergence entre les deux distributions.

Condition d'arrêt Pour chacune des fonctions objectif, nous utilisons la fonction *Adam* comme condition d'arrêt : elle contrôle le taux d'apprentissage initialisé à 0,0001 pour l'optimisation stochastique. De plus, afin d'éviter les erreurs de calcul dans les limites, nous fixons les valeurs de sortie (0,1) à ($1e^{-6}$, $1\text{-}1e^{-6}$).

Évaluation des performances et méta-paramètres Nous souhaitons évaluer les performances de notre modèle optique pour construire un vecteur de n-grammes de caractères. Pour cela nous calculons le rappel, la précision, l'exactitude et le F-mesure comme défini dans l'équation 5.1. Nous prenons le parti de mesurer si les n-grammes composant le mot sont présents en définissant un seuil d'acceptation, sans vérifier si la fréquence prédite des n-grammes est proche de la vérité terrain. Nous définissons les n-grammes correctement identifiés dans le mot vecteur comme les vrais positifs (VP), alors que les n-grammes hors du mot et non détectés sont les vrais négatifs (VN). Les n-grammes initialement inclus dans le vecteur de mot et non ajoutés par le modèle sont les faux négatifs (FN). Enfin, les n-grammes à l'origine non inclus dans la décomposition du mot, mais ajoutés par le modèle sont les faux positifs (FP). L'ensemble de ces informations est résumé dans la matrice de confusion proposée dans la table 5.2. À partir de ces définitions, nous fournissons la description des mesures de performances :

$$\begin{aligned} \text{Rappel} = \frac{VP}{VP+FN}, \ \text{Précision} = \frac{VP}{VP+FP}, \\ \text{Exactitude} = \frac{VP+VN}{VP+VN+FP+FN}, \\ \text{F-Mesure} = 2 \times \frac{Rappel \times Précision}{Rappel + Précision}. \end{aligned} \tag{5.1}$$

Pour réaliser l'apprentissage, nous créons des sous-ensembles de données (*mini-batch*) de taille 10, pour pouvoir accélérer le processus. L'ensemble des méta-paramètres sont mis à jour à chaque itération. Pour prévenir le sur-apprentissage du modèle, nous mettons en place la technique de l'*early stopping*. Cette méthode utilise le coût de la fonction de perte sur la base de validation du système. Si ce dernier ne décroît pas durant 5 itérations successives, l'apprentissage est arrêté.

		Réelle		
		Vrai	Faux	Total
Prédiction	Vrai	N-grammes correctement attribués au mot VP	N-grammes attribués par erreur au mot FP	Précision $= \frac{VP}{VP+FP}$
	Faux	N-grammes exclus par erreur du mot FN	N-grammes correctement exclus du mot VN	-
	Total	Rappel $= \frac{VP}{VP+FN}$	-	N

TABLE 5.2 – Matrice de confusion pour classifier les résultats obtenus

5.3.2 Résultats et observations

Dans cette section, nous présentons les expériences et leurs résultats pour définir le modèle optique. La première partie vise à définir si la couche pyramidale est plus adaptée que les masques comme modèle d'attention artificiel. La seconde partie compare les différentes configurations des fonctions d'activation, des fonctions de coût, de la taille des n-grammes de caractères et enfin de la taille de la dernière couche FC9. Les expériences sont menées à la fois avec un apprentissage sur ESP et un apprentissage par transfert de connaissances avec GW et RM.

Définition du cas utilisateur Comme nous n'avons pas d'images de mots isolés de la Comédie-Italienne, nous définissons un cas utilisateur pour expérimenter notre approche avec l'ensemble de données ESP comme données inconnues à transcrire. Soit I_{ESP}, un ensemble d'images de ESP considéré comme une nouvelle ressource sans vérité de terrain que nous voulons annoter. Le vocabulaire principal de cette base de données historique est composé d'entités nommées et la langue utilisée est le catalan avec un alphabet latin. Les bases de données GW et RM sont utilisées pour réaliser l'apprentissage du système. Nous utilisons également la base d'apprentissage de ESP pour évaluer localement les performances du système, c'est-à-dire définir si les résultats obtenus sont liés au modèle ou à l'apprentissage par transfert de connaissances.

Système pyramidale et attention artificielle Les deux stratégies mises en œvre fonctionnent pour des images de mots isolés. Cependant, à long terme, nous souhaitons travailler sur des images plus grandes comme des lignes de titres. Il faut donc être capable d'identifier les différents mots et leur position dans l'image. Pour cela, un modèle d'attention automatique (et non artificiel) devra être mis en place pour identifier les mots, mais également pour générer l'ensemble des 6 masques automatiquement. Cela revient à un modèle de segmentation d'image comme (Badrinarayanan et al. 2015).

Comparaison de différentes configurations La table 5.3 présente les résultats de nos expériences pour le modèle optique. Le modèle que nous avons utilisé comme base de nos expériences B0 et E0, est construit avec la configuration de la couche de sortie SIGOFF, la fonction objectif Entropie et l'ensemble des n-grammes de caractères en sortie. Nous avons fait varier chacun de ses paramètres pour pouvoir mesurer l'impact de chacun sur la qualité du vecteur de n-grammes de caractères. Cependant, toutes les combinaisons n'ont pas été réalisées. La première partie des résultats permet de définir la configuration du système donnant les

meilleurs rappels, précision et F-mesure en utilisant ESP pour entraîner et tester le système. La deuxième partie des résultats est obtenue en utilisant l'apprentissage par transfert de connaissances avec GW et RM sur l'ensemble de données ESP. Nous réalisons l'apprentissage du réseau avec les ressources GW et RM combinées puis les tests sont réalisés sur l'ensemble de validation de ESP. Nous présentons ici les expériences qui ont été menées avec un seuil d'évaluation fixé à 0,5 comme la valeur du décalage ajoutée pour calculer les différentes mesures d'évaluation et ainsi identifier les n-grammes acceptés. Nous souhaitons obtenir un vecteur de n-grammes de caractères possédant le plus de n-grammes du mot d'origine. Nous nous focaliserons principalement sur le rappel et la F-mesure pour cela.

TABLE 5.3 – Dans la partie supérieure, l'évaluation de la meilleure configuration sur l'ensemble de données d'apprentissage ESP. Dans la partie inférieure, les résultats obtenus sur l'apprentissage par transfert de connaissances avec les ensembles d'apprentissage de GW et RIMES. Tous les résultats sont calculés sur les données de validation de ESP. (%)

Expe. Id	Couche FC9	Config	Fonct. coût	Nb masques	N-grammes caractères	Rappel %	Précision %	F-Mesure %	Exactitude %
B0	1024	SIGOFF	Bin.	1	1,2,3	29,03	47,28	35,84	99,89
B1	1024	SIGOFF	Bin.	6	1	**89,61**	66,48	76,33	98,33
B2	1024	SIGOFF	Bin.	6	1,2	79,03	61,85	69,40	99,68
B3	1024	SIGOFF	Bin.	6	1,2,3	72,40	58,16	64,50	99,91
B4	2048	SIG	Bin.	6	1,2,3	74,39	**93,16**	**82,72**	99,96
B5	2048	SIGOFF	Kull	6	1,2,3	11,72	55,68	19,10	99,90
B6	2048	SIGOFF	Bin	6	1,2,3	65,89	92,34	76,91	99,96
E0	1024	SIGOFF	Bin.	6	1	35,48	29,09	**31,97**	95,48
E1	1024	SIGOFF	Bin.	6	1,2	34,14	27,13	22,51	99,16
E2	1024	SIGOFF	Bin.	6	1,2,3	18,77	32,43	23,78	99,88
E3	1024	SIGOFF	MSE	6	1,2,3	10,73	34,80	16,27	99,88
E4	1024	THR	Bin.	6	1,2,3	**63,40**	4,29	8,04	98,52
E5	1024	THR	MSE	6	1,2,3	29,46	11,02	15,99	99,69
E6	2048	SIGOFF	Bin.	6	1,2,3	10,58	**45,16**	17,15	99,90
E7	2048	SIG	MSE	6	1,2,3	9,53	37,11	15,07	99,89
E8	2048	THR	Bin.	6	1,2,3	59,32	4,77	8,83	98,76
E9	2048	THR	MSE	6	1,2,3	28,33	12,40	17,19	99,72

Premièrement, nous nous concentrons sur l'impact de la configuration sur les résultats grâce aux expériences B_i. Les expériences B0 et B3 montrent que cibler certaines zones, grâce à 6 masques sur les caractéristiques qui ont été extraites par le système de convolution, est plus efficace que de considérer les caractéristiques globalement sur tout le mot. De plus, les expériences B1, B2 et B3 montrent que la longueur des n-grammes de caractères, et donc la quantité des n-grammes de caractères, a un effet notable. La différence de rappel entre B1 et B3 est liée au nombre de sorties disponibles. En effet, plus on a de sorties possibles (B3), moins le rappel est bon. Pour l'expérience B1, le nombre de sorties possible avec les unigrammes (et bigrammes incluant les débuts et fins de mots avec un caractère) est limité à 199, ce qui facilite l'identification et donne les meilleurs taux de rappel,

précision et F-mesure par rapport à B2 (seulement 1 883 n-grammes de caractères) et B3. Nous en concluons que plus le cas est simple, plus le taux de rappel est élevé. Une analyse plus poussée des prédictions réalisées montre que le modèle B1 prédit 3 fois plus d'unigrammes que ceux attendus, ce qui explique le fort taux de rappel obtenu. Quant à B3, le modèle prédit 7 fois plus de n-grammes de caractères, mais sans pour autant identifier ceux attendus ce qui explique que les taux de rappel et précision soient tous les deux moins bons.

Afin de mesurer l'efficacité du décalage de la fréquence par rapport à une cible binaire (1 pour les n-grammes présents, 0 sinon), nous comparons les expériences B4 et B6. L'encodage de chaque sortie avec une information binaire (SIG) est plus efficace que la probabilité de SIGOFF. B6 perd seulement 8,5% sur le rappel par rapport à B4 tandis que la précision reste identique. Cette faible différence entre les deux expériences est un atout, car pour le décodeur, la notion de quantité des n-grammes de caractères est nécessaire pour reconstruire le mot.

Concernant la fonction de coût, l'entropie utilisée dans B6 fournit un rappel largement meilleur de plus de 55%, de même pour la précision et la F-mesure qui sont largement supérieures, par rapport à l'utilisation de la fonction de Kullback-Liebler (B5). Ces résultats vont à l'encontre de l'intuition que l'on peut avoir en analysant le détail des fonctions puisque l'entropie est normalement utilisée pour une cible binaire et non une fréquence.

En ce qui concerne ces résultats, la meilleure configuration est opérée par l'entropie binaire comme fonction de coût, en utilisant la fréquence normalisée avec le décalage et les caractéristiques filtrées avec 6 masques basés sur l'image d'entrée. Avec un rappel et une F-mesure supérieurs à 65%, nous montrons qu'il est possible de construire un vecteur n-grammes de caractères pour représenter une image de mot.

Nous nous concentrons sur les expériences par apprentissage par transfert de connaissances de E_i. Le premier constat est que pour des configurations équivalentes dans la partie B_i, les résultats perdent en moyenne 50% pour le rappel, 31% pour la précision et 40% pour la F-mesure.

La configuration THR utilisée pour l'expérience E4 fait énormément chuter la précision et la F-mesure. Bien qu'un fort rappel soit préférable pour notre modèle, les taux de précision et de F-mesure ne doivent pas être quasi-nuls.

La fonction de coût MSE a une tendance similaire selon la configuration utilisée pour la construction du vecteur. Avec la configuration SIGOFF dans E3, le rappel chute très légèrement par rapport à E2. Tandis qu'avec THR (voir E4 et E5), le rappel perd 30% et la précision diminue également de 7%. Les meilleures F-mesures

sont obtenues avec les unigrammes, bigrammes et trigrammes de caractères pour une configuration la plus simple c'est-à-dire en utilisant la configuration SIGOFF avec sa fonction de coût, et seulement 1 024 neurones sur la couche FC9.

Pour finir, l'ensemble des expériences B_i et E_i ne nous ont pas permis de prendre une décision sur la meilleure structure à envisager pour la couche FC9. Dans le cadre des expériences de type B_i sur ESP, les meilleurs résultats tendent vers une structure à 2 048 neurones. Avec plus de neurones dans cette dernière couche, le modèle se spécialise aux données de ESP. Tandis qu'avec les expériences E_i, une structure plus petite composée de 1 024 neurones avec l'ensemble des n-grammes de caractères semble plus adaptée à l'apprentissage par transfert de connaissances. Les expériences prouvent que les architectures plus simples favorisent la généralisation, ce qui est très important pour notre étude.

Seuil d'évaluation Dans certaines des expériences précédentes, un décalage positif de 0,5 a été ajouté pour augmenter la différence entre les sorties cibles des n-grammes de caractères présents dans l'image de mot de ceux non-présents. Pour évaluer la composition du vecteur, nous avions fixé le seuil d'évaluation à ce taux de décalage de 0,5. Nous considérions un n-gramme faisant partie du vecteur s'il avait une sortie au-dessus de ce seuil. Par exemple, pour le n-gramme "[So" de notre exemple "Sophie" de Comédie-Italienne, la valeur cible est de $0,5 + \frac{1}{19} = 0,55$. Pour des mots plus longs, les valeurs cibles sont très proches de 0,5. L'ensemble de test de ESP contient en moyenne 12 n-grammes par mot. Cependant avec ce seuil, le modèle B6, par exemple, produit des vecteurs de seulement 4,9 n-grammes.

Pour pouvoir constater l'impact de ce seuil sur les résultats, nous réalisons un ensemble d'expériences dans lesquelles nous le faisons varier. Nous utilisons le modèle basé sur l'architecture présentée en figure 5.3 avec 2028 neurones pour la couche FC9. Pour constituer le vecteur, la couche SIGOFF est utilisée avec sa fonction de perte, l'entropie binaire. Les figures 5.6 et 5.7 présentent les courbes de rappel, la précision et la F-mesure pour sept valeurs différentes de seuil : {0,01 ; 0,05 ; 0,1 ; 0,2 ; 0,3 ; 0,4 ; 0,5} sur différentes ressources d'apprentissage.

Nous nous plaçons dans un premier temps dans un cadre d'apprentissage de modèle classique avec ESP comme ressource d'apprentissage avec la figure 5.6. Pour obtenir un système performant c'est-à-dire un vecteur dans lequel nous avons un maximum de n-grammes de caractères du mot présent, nous considérons le rappel. Sa courbe présente des résultats supérieurs à 75% pour des seuils inférieurs à 0,2. Cependant pour un vecteur optimisé, c'est-à-dire qui accepte un maximum de n-grammes pertinents tout en rejetant les non-pertinents, la F-mesure fournit un résultat similaire pour un seuil défini à 0,2 et 0,4.

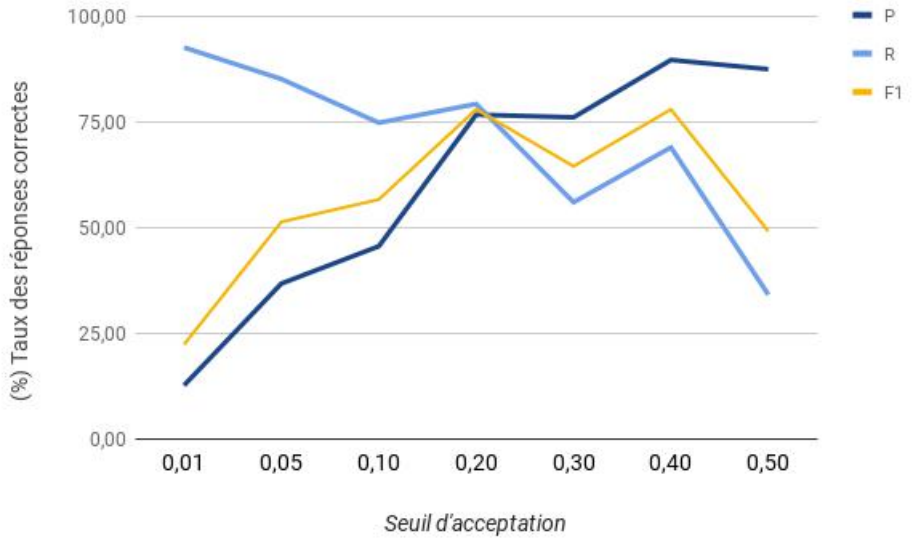


FIGURE 5.6 – Évaluation du seuil d'acceptation des n-grammes de caractères sur la ressource validation d'ESP (l'entraînement a été réalisé sur la partie apprentissage d'ESP).

Grâce à la figure 5.7, nous pouvons vérifier si le comportement de l'apprentissage par transfert de connaissances suit la même variation que lors d'un apprentissage classique. Par comparaison, il apparaît que l'ensemble des résultats perd 30% dans le cadre de l'apprentissage par transfert de connaissances. Malgré tout, les valeurs de seuil remarquables restent les mêmes pour la F-mesure ou le rappel. Finalement, le seuil de 0,2 permet de construire un vecteur plus pertinent.

La table 5.4 qui fournit la taille moyenne des vecteurs obtenus lors de l'apprentissage classique sur ESP. Elle montre que le seuil de 0,2 crée des vecteurs de la taille attendue c'est-à-dire 12 n-grammes de caractères par image de mot.

Convolution supplémentaire Les résultats obtenus peuvent être encore optimisés. Nous avons défini la structure, les fonctions de coût et d'activation ainsi que le seuil permettant de construire un vecteur de n-grammes de caractères le plus proche possible de la vérité. Nous proposons une dernière expérimentation (voir table 5.5) pour tenter d'améliorer encore les résultats en ajoutant une huitième convolution (Conv8) de la même taille que la précédente c'est-à-dire 512. La F-mesure obtenue avec seulement 7 convolutions et un seuil de 0,2 est similaire à celle obtenue avec 8 convolutions et le seuil à valeur du décalage. Le rappel perd 14%, mais reste au-dessus des 50%, car la longueur moyenne des vecteurs créés est

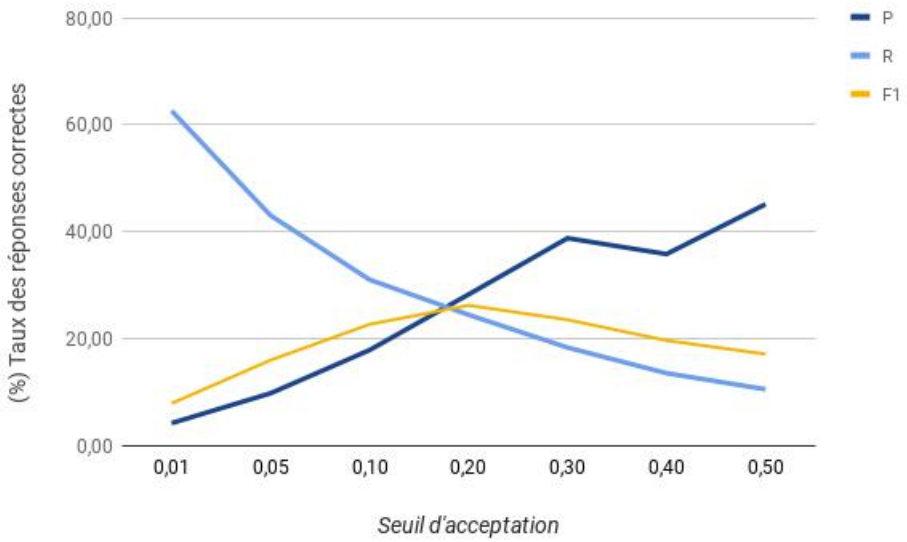


FIGURE 5.7 – Évaluation du seuil d'acceptation des n-grammes de caractères sur la ressource ESP en validation (l'entraînement a été réalisé sur la partie apprentissage de GW et RM).

TABLE 5.4 – Taille moyenne des vecteurs de n-grammes de caractères suivant le seuil d'évaluation utilisé (la vérité terrain comporte en moyenne 12 n-grammes dans les vecteurs).

Seuil	Nombre moyen de n-grammes
0,01	89,7
0,05	28,52
0,1	20,19
0,2	12,72
0,3	9,07
0,4	9,5
0,5	4,9

de 8,8 n-grammes. Finalement, nous observons le meilleur score de la F-mesure avec le seuil fixé à 0,2 et 8 convolutions avec une augmentation de 8 % par rapport au score obtenu avec seulement 7 convolutions. Nous pouvons en conclure que dans un cas d'apprentissage classique c'est-à-dire avec des données provenant de la même ressource pour l'apprentissage et l'évaluation du système, le modèle le plus performant est le modèle le plus grand, avec 8 convolutions et le seuil de 0,2 que nous avions estimé précédemment. Dans le cadre d'apprentissage par transfert de connaissances, il est moins sûr que ce modèle soit le meilleur, en effet, la convolution supplémentaire peut favoriser la reconnaissance spécifiquement pour les données ESP utilisées dans ce cas précis en apprentissage et validation. Nous faisons le choix de conserver le modèle à 7 convolutions pour une partie des expériences à venir afin de préserver la généralisation du système.

TABLE 5.5 – Résultats obtenus pour un apprentissage et une validation sur ESP avec 7 et 8 convolutions successives pour extraire les caractéristiques.

# Convolution	Seuil	Rappel %	Précision %	F-mesure %
Conv 7	0,2	79,17	76,65	77,89
	0,5	34,13	87,38	49,09
Conv 8	0,2	87,38	82,36	**84,8**
	0,5	65,88	92,34	76,91

5.3.3 Conclusion

Dans cette section, nous avons éprouvé les différentes solutions envisagées pour construire la partie modèle optique. Nous avons privilégié une architecture utilisant des masques artificiels permettant de simuler un modèle d'attention sur les caractéristiques extraites par les convolutions. L'ensemble des expérimentations menées nous permettent de créer une cartographie des meilleures configurations en fonction de la tâche effectuée. En effet, les configurations les plus performantes en apprentissage par transfert de connaissances ne sont pas les mêmes que celles pour un apprentissage classique. De plus, les expériences utilisant différents n-grammes montrent que plus il y a de n-grammes à identifier parmi un large choix, plus il est difficile d'obtenir de bons résultats. Cependant, il nous paraît important de conserver l'ensemble de n-grammes définis pour que le décodeur puisse reconstruire le mot.

Lors de l'ensemble de ces expérimentations, les poids de chaque réseau ont

été sauvegardés pour pouvoir initialiser les poids du modèle encodeur-décodeur complet. Nous avons conservé un sous-ensemble des meilleures configurations parmi les E_i, essentiellement pour évaluer ensuite le modèle encodeur-décodeur complet.

5.4 Modélisation du décodeur

Nous venons de présenter les différentes architectures et paramètres pour le modèle optique. Nous nous focalisons dans cette section sur la réalisation du décodeur.

La reconnaissance d'écriture est une tâche très similaire à la génération de description d'images. Dans les deux cas, nous avons un réseau qui extrait un ensemble de caractéristiques d'une image avec l'aide d'un modèle d'attention qui isole des parties de l'image. Cela permet au réseau d'identifier les objets un à un et de finalement générer une phrase descriptive de l'image. Pour cela, le modèle donne une probabilité pour chaque mot du vocabulaire, à chaque instant. Dans notre étude, un modèle de ce type pourrait être une option envisageable à condition que le vocabulaire utilisé pour la sortie couvre pleinement les éléments que l'on cherche à identifier. Dans le cadre d'une approche utilisant l'apprentissage par transfert de connaissances à l'aide de ressources multi-domaines et multi-langues, cette condition paraît difficile à remplir tant le nombre de mots hors vocabulaire est grand. Nous avons par conséquent écarté cette solution.

Nous avons opté pour une solution générant des caractères un à un, jusqu'au symbole fin de mot, à partir des n-grammes de caractères présents dans le vecteur d'entrée. La figure 5.8 présente le modèle mis en place pour cette partie. Son architecture est détaillée dans la sous-section suivante puis les résultats des expériences menées sont présentés.

5.4.1 Architecture du modèle

Les modèles pour réaliser de la TA se basent sur (Sutskever et al. 2014 ; Cho et al. 2014a) comme point de départ. Leur composant type décodeur possède très peu de couches. Par exemple, l'architecture (Sutskever et al. 2014) se compose d'une couche d'entrée de taille conséquente puisqu'elle représente les 160 000 mots du vocabulaire d'entrée, d'une couche de plongement de mots pré-entraînée de 1 000 neurones, suivi de 4 couches de LSTM de 1 000 cellules et enfin d'une couche de sortie représentant les 80 000 mots du vocabulaire. Pour construire notre modèle,

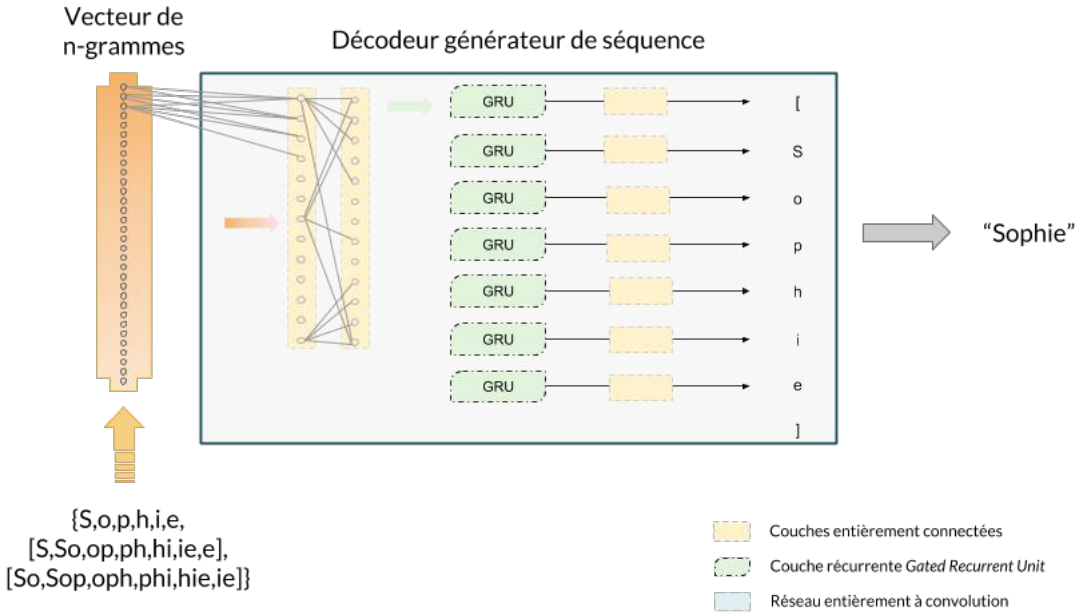


FIGURE 5.8 – Fonctionnement et architecture du décodeur générateur de séquences de caractères

nous conservons le rôle de chaque couche. Voici l'implémentation réalisée pour générer la séquence de caractères à partir du vecteur de n-grammes :

- une couche entièrement connectée avec une activation de type ReLU pour réaliser le plongement de n-grammes de caractères ;
- une couche récurrente de type GRU ;
- une dernière couche entièrement connectée avec une fonction d'activation de type *Softmax*.

Cette architecture volontairement simple pourrait l'être encore plus si nous utilisions une couche de plongement lexical pré-entraînée. Mais à notre connaissance, il n'existe pas de plongements lexicaux multilingues pour les n-grammes disponibles. Pour obtenir une séquence, nous utilisons une couche récurrente qui va générer de la temporalité. La dernière couche doit fournir un caractère jusqu'à ce que le symbole de fin de mot soit émis.

Dans les modèles encodeur-décodeur, notamment utilisés en traduction, la partie décodeur utilise un réseau bi-directionnel pour prendre en compte toutes les informations contenues dans une phrase à travers une matrice. Or, nous utilisons un vecteur comme entrée ce qui signifie que l'information temporelle spécifiant la position d'un n-gramme par rapport à un autre a disparu. Un réseau bidirectionnel ne sera donc pas utile dans notre cas. La structure du décodeur comprend ainsi 1 024

neurones dans la première couche entière connectée pour extraire les caractéristiques, suivi de 500 neurones cachés dans la couche GRU et enfin 79 neurones avec *Softmax* en fonction d'activation. Pour générer les séquences de sorties, les 79 neurones représentent toutes les lettres minuscules et majuscules, les chiffres, les symboles de ponctuation dont l'espace et enfin les symboles de début et fin de mot. Un dernier neurone est ajouté pour permettre au réseau de ne plus répondre de caractères après la fin du mot.

5.4.2 Hyper-paramètres et condition d'apprentissage

Hyper-paramètres Le taux d'apprentissage fixé à 0,0001 est géré automatiquement grâce à la fonction Adam. À partir du vecteur en entrée, le réseau doit générer une séquence. Nous avons défini arbitrairement une limite sur la taille des séquences à générer pour aider le modèle. Cela se traduit par une répétition de la couche de plongement, 50 fois. Cette valeur peut sembler grande pour la génération de mot, mais pour générer les titres de pièces de la Comédie-Italienne, cela est approprié. En effet, parmi les ressources que nous utilisons, la longueur maximum des mots est de 22 caractères ("ballettodivertissement" dans GCI) et les lignes RCI à transcrire (qui restent notre objectif) ne dépassent pas non plus une séquence de plus de 50 caractères.

Condition d'apprentissage Comme la taille de nos ressources varie de 660 à environ 25 000 mots, nous avons entraîné le décodeur avec une ou plusieurs ressources hormis pour GW (comme la taille est trop petite, nous combinons systématiquement cette ressource à d'autres). Pour éviter le sur-apprentissage dans notre réseau, nous utilisons la méthode d'arrêt prématuré qui consiste à stopper le réseau quand, au bout de cinq itérations, la fonction coût sur la base de validation ne décroit plus.

Ressource supplémentaire Traditionnellement, en traduction, pour évaluer une méthode multilingue ou combler un manque de données, Wikipédia est la ressource la plus utilisée. Elle fournit une très grande quantité de données multilingues, mais les textes sont écrits dans un style moderne. Nous travaillons sur des données historiques et avec un vocabulaire fermé, mais il est intéressant de pouvoir observer l'impact de cette ressource d'apprentissage sur les autres ressources. Nous utilisons Wikipédia français, utilisée et distribuée par (Bojanowski et al. 2017). Cette ressource contient tous les mots qui ont une fréquence supérieure à 5 dans Wikipédia. Pour pouvoir rester dans un domaine comparable avec les ressources que nous

avons déjà, nous en sélectionnons aléatoirement 30 000 mots (Wiki 30k) et 300 000 mots (Wiki 300k).

Évaluation Nous évaluons notre système grâce à un taux de reconnaissance sur les caractères (TRC) et sur les mots (TRM). Le TRC est défini par

$$\text{TRC} = \max(0, \frac{(N - (Ins + Subs + Dels))}{N}) \times 100$$

où N représente le nombre de caractères dans le mot de référence, $Subs$ le nombre de caractères substitués, $Dels$ le nombre de caractères supprimés et Ins le nombre de caractères insérés. Le TRM correspond au rappel, c'est-à-dire au nombre de mots correctement reconnus par rapport au nombre de mots dans l'ensemble de référence.

Dans un premier temps, nous avons calculé ces taux avec différentes options sur les ensembles de validation pour définir les meilleures combinaisons de ressources suivant le langage et la période ciblée. Dans un second temps, ces mêmes expérimentations ont été réalisées sur les ensembles de tests de chaque ressource. Les options utilisées sont les suivantes : i) avec et sans dictionnaire pour aider au décodage de la séquence et ii) avec et sans majuscule pendant le décodage.

Le dictionnaire est construit sur le vocabulaire des ensembles d'apprentissages et de validation de l'ensemble des ressources hormis Wikipédia. Cela nous donne un dictionnaire avec 39 051 entrées. Le dictionnaire reste une aide pour le décodage, mais il ne comporte pas l'ensemble des mots à décoder. Pour connaître la limite de son utilisation, nous calculons la couverture lexicale fournie par le dictionnaire par rapport aux ensembles de tests indiquée dans la colonne "% Couv. lexicale dict.". Cela nous donne une borne haute pour le taux de reconnaissance de mots qui peut être atteint. En effet, certains mots peuvent être communs à plusieurs ressources et ne pas appartenir à l'ensemble d'apprentissage. Cette couverture lexicale correspond au nombre de mots communs entre le dictionnaire et l'ensemble de test, divisé par la taille du vocabulaire de l'ensemble de test. Nous calculons également la couverture lexicale entre l'ensemble d'apprentissage et de test, de la même façon que la couverture lexicale par rapport au dictionnaire afin de montrer la différence entre le vocabulaire utilisé en apprentissage et celui que l'on tente de générer en test.

5.4.3 Résultats et observations

La table 5.6 montre les résultats obtenus pour la génération de séquences. Nous avons réalisé trois types d'expériences :

1. avec la même ressource pour l'apprentissage et le test ;
2. en ajoutant d'autres ressources pour l'apprentissage ;
3. en utilisant uniquement des ressources différentes de l'ensemble de test, pour l'apprentissage, ce qui correspond à l'apprentissage par transfert de connaissances.

Même si le but de notre approche est bien de pouvoir décoder les mots de la Comédie-Italienne, nous présentons aussi les résultats obtenus sur RM et ESP. Cela nous semble intéressant de pouvoir observer si la méthode est applicable pour différents types de ressources.

Choix de l'ordre des n-grammes Pour commencer, nous utilisons uniquement les unigrammes pour représenter un mot avec la même ressource pour l'apprentissage et le test. Nous constatons sur les trois ressources différentes que les résultats sont meilleurs lorsque l'on considère les 1,2,3-grammes de caractères. Sur GCI, le TRM augmente de 74 % sans l'utilisation du dictionnaire, tout en dépassant la couverture lexicale. Notons également que seuls 3 % des caractères sont mal reconnus sur l'ensemble de test. Ces résultats corroborent les études utilisant les trigrammes comme (Vania et Lopez 2017) ainsi que celles utilisant les n-grammes de caractères pour leur robustesse aux mots hors-vocabulaire comme (Huang et al. 2013a). Pour la suite des expériences, nous utilisons uniquement les vecteurs de 1,2,3-grammes.

Prise en compte des majuscules et du dictionnaire Sur l'ensemble des expériences, nous obtenons globalement les mêmes résultats avec ou sans majuscules : ± 0.07 % pour le TRC, ± 1 % pour le TRM et ± 1.05 % avec le dictionnaire. Les erreurs de caractères du système ne sont donc pas uniquement des minuscules prédites en majuscules. L'utilisation du dictionnaire fait chuter les performances du décodeur. Nous remarquons que lorsque la couverture lexicale est supérieure à 20 %, le TRM reste inférieur à celle-ci. Le dictionnaire induit en erreur le décodeur quand il génère une séquence correcte et que le mot correspondant n'appartient pas au dictionnaire. Cependant, il est également capable d'aider le décodeur à se rapprocher du mot de référence même si la forme exacte n'est pas contenue dans le dictionnaire, et qu'en plus, les données d'apprentissage et de test utilisées sont d'époques et de langues différentes. Seulement, ces cas sont trop rares pour améliorer le TRC.

Test sur RCI Les résultats obtenus pour les TRC et TRM, lorsque GCI est combiné avec d'autres ressources, sont très similaires aux résultats obtenus sur GCI

Test	Apprentissage	Expe. Id	N-grams	% Couv. lexicale dict.	% Couv. lexicale	Validation			Test		
						TRC	TRM	TRM dict.	TRC	TRM	TRM dict.
RCI	GCI	DCI1	1	73,44	65,57	69,11	13,05	14,65	69,27	14,54	10,83
	GCI	DCI2	1,2,3			**95,85**	**77,83**	**66,47**	**97,10**	**86,22**	**39,30**
	GCI+RM	DCI3	1,2,3	73,44	67,53	95,97	78,78	66,17	**97,27**	**86,57**	**39,23**
	GCI+ESP	DCI4	1,2,3		65,83	**96,49**	**85,87**	**67,22**	96,96	81,46	39,09
	GCI+ESP +GW+RM	DCI5	1,2,3		67,65	95,99	77,86	66,37	95,85	79,65	38,25
	RM	DCI6	1,2,3	73,44	14,52	74,69	19,49	23,32	79,70	30,42	17,27
	RM+ESP+GW	DCI7	1,2,3		23,39	78,11	25,36	28,44	83,68	40,21	23,99
	Wiki 30k	DCI8	1,2,3		0.0	**83,67**	**41,18**	**42,75**	**87,32**	**41,40**	**25,24**
RM	RM	DR9	1	86,29	75,09	82,66	37,83	27,58	83,97	43,07	28,49
	RM	DR10	1,2,3			**93,95**	**79,93**	**37,45**	**94,72**	**79,50**	**37,78**
	GCI+RM	DR11	1,2,3	86,29	83.83	**97,32**	**89,04**	**39,94**	**98,25**	**92,0**	**40.49**
	GCI+ESP +GW+RM	DR12	1,2,3		83,95	96,8	86,18	39,43	96,22	80,73	39,14
	GCI	DR13	1,2,3	86,29	58,55	**93,93**	**75,48**	**36,56**	**95,51**	**81,53**	**38,58**
	GCI+ESP	DR14	1,2,3		59,04	93,89	74,33	36,05	95,46	80,61	38,15
	Wiki 30k	DR15	1,2,3		0,0	87,04	65,99	33,31	90,36	67,57	35,20
	Wiki 300k	DR16	1,2,3		0,0	93,36	83,69	36,70	93,44	83,88	37,60
ESP	GCI+ESP	DE17	1	89,42	85,94	85,46	44,43	37,87	83,36	37,61	37,94
	GCI+ESP	DE18	1,2			92,00	62,31	59,49	95,21	71,1	53,81
	GCI+ESP	DE19	1,2,3			**98,25**	**91,61**	**62,92**	**98,57**	**91,11**	56,51
	GCI+ESP +GW+RM	DE20	1,2,3		86,10	98,14	90,48	61,94	98,40	90,79	**57,62**
	GCI	DE21	1,2,3	89,42	15,96	88,18	54,35	45,48	91,68	**65,87**	44,76
	RM	DE22	1,2,3		7,27	67,82	14,03	8,87	72,83	18,25	12,70
	GCI+RM	DE23	1,2,3		17,37	87,4	41,77	**51,61**	92,05	64,13	**46,51**
	GCI+RM+GW	DE24	1,2,3		17,69	**88,77**	**57,1**	44,35	91,68	64,60	44,60
	Wiki 30k	DE25	1,2,3		0,0	78,89	27,26	24,52	84,52	34,28	32,38
	Wiki 300k	DE26	1,2,3		0,0	91,96	54,29	44,60	**92,23**	56,19	46,03

TABLE 5.6 – Résultats pour la génération de séquence expérimentant l'apprentissage par transfert de connaissances : les TRC et TRM sont calculés avec ou sans l'utilisation du dictionnaire sur les différentes ressources. Pour chaque ensemble de test et de validation, la meilleure valeur obtenue est indiquée en gras pour chacun des 3 types d'expériences (en dehors de celles réalisées avec Wiki 300k).

seul (expérience DCI2 *vs.* expériences DCI3, DCI4 et DCI5). Dans ce cas précis, l'augmentation de la quantité de données n'a pas un impact flagrant. Notons quand même que les meilleurs résultats sont obtenus en utilisant toutes les ressources en apprentissage. Dans le cas de l'apprentissage par transfert par rapport au domaine, la couverture lexicale est très basse, puisqu'elle est autour de 15 %. Cependant, le décodeur est capable d'atteindre 30,42 % de TRM en utilisant uniquement RM, qui est en français contemporain, et 41,40 % en utilisant Wikipédia. Les résultats avec Wikipédia sont similaires à ceux obtenus en utilisant toutes les ressources avec une quantité inférieure de données. Finalement, chercher de nouvelles ressources encore inexploitées sur la Comédie-Italienne est une approche intéressante : cela nous permet d'avoir un TRM supérieur de 20 % à la couverture lexicale. Parmi ces mots inconnus, mais bien reconnus, nous retrouvons des abréviations telles que « arleq. » au lieu d'« Arlequin ».

Test sur RIMES Les résultats avec RM sont intéressants, car c'est la seule ressource que nous utilisons qui est en français contemporain. Dans le cadre de l'apprentissage par transfert de connaissances, sans RM dans les ressources d'apprentissage, les TRC et TRM atteignent les résultats obtenus sur RM seul. De plus, nous constatons qu'ils dépassent largement la couverture lexicale de 20 %. Lorsque nous travaillons avec des données historiques appliquées sur des données modernes, mais partageant la même langue, la couverture lexicale est plus élevée que lorsque l'on utilise RM sur GCI. Ainsi, l'orthographe historique est plus facilement applicable sur du moderne pour le décodeur.

Test sur Les Esposalles Le vocabulaire de Les Esposalles est principalement construit à partir d'entités nommées. Ceci explique la couverture lexicale faible, voire nulle, dans le cadre de l'apprentissage par transfert de connaissances. Malgré cela, le TRC est élevé puisqu'il augmente de 70 % à 90 %, et le TRM dépasse la couverture lexicale jusqu'à 57,1%. Nous n'avons pas expérimenté l'utilisation seule de la base d'apprentissage de référence de cette ressource, car elle est trop petite.

Analyse des erreurs La table 5.7 montre quelques erreurs récurrentes que nous avons pu constater. Parmi les erreurs observées, nous constatons que les mots ayant des caractères répétés, comme « cavalcade » et « clemence », posent plus de difficultés au système pour générer les caractères qui s'intercalent avec le caractère répété : il propose « cacaadade » et « ccceeene », respectivement. Cela représente environ 5 % des erreurs constatées dans les expériences sur RCI. Une autre erreur commune est la permutation entre deux caractères comme avec « suitte » qui double un caractère à la place d'un autre. Cette erreur est la plus commune, elle couvre

79 % des erreurs observées. Un dernier exemple avec « [ollat » pour « Soldat » qui apparaît lorsque le décodeur prédit deux symboles début de mot consécutifs, le second remplaçant le premier caractère du mot. Suivant la taille de la ressource utilisée pour l'apprentissage, ce type d'erreur représente entre 7 % et 25 % des erreurs, respectivement avec GCI, et RM seul.

Type d'erreur	Expe. Id	Mot d'origine	Mot reconstitué
Caractère multiplié	DCI4 DCI3	cavalcade clemence	cacaadade ccceeene
Caractère interverti	DCI6	suitte	usitte
Caractère de début	DCI5 DCI6	[diverstissemens] Soldat	ddevvestissemens] [ollat

TABLE 5.7 – Exemples d'erreurs réalisées par le décodeur.

5.4.4 Amélioration du modèle

Le modèle présenté montre de bons résultats pour l'apprentissage par transfert de connaissances. Le vecteur fourni en entrée du modèle est idéalement formé puisque les n-grammes de caractères non présents dans le mot sont à zéro tandis que les autres fournissent leur fréquence. Cependant, la sortie du modèle optique n'est pas si parfaite. La valeur des n-grammes de caractères n'est jamais réellement nulle. Nous souhaitons mettre le modèle dans des conditions plus réalistes en réalisant un apprentissage à partir d'un signal bruité pour le décodeur, pour qu'il soit plus robuste.

Entrée bruitée Nous expérimentons deux approches différentes pour bruiter le signal.

Fonction Gaussienne Nous créons un bruit artificiel grâce à une fonction aléatoire gaussienne. Nous définissons la fonction avec une déviation standard de 0,1 et une moyenne de 0 qui donne un vecteur qui est additionné à celui des n-grammes de caractères avant la couche de plongement.

Couche bruité Il existe dans la librairie *Lasagne*, que nous utilisons, une couche nommée "*Gaussian noise layer*" (Couche de bruit gaussien). Elle ajoute directement le bruit gaussien à l'objet qui lui est passé en entrée suivant la déviation fournie. La moyenne n'est pas directement paramétrable : elle

est par défaut à 0. Nous plaçons cette couche entre la couche d'entrée du modèle et celle du plongement.

Pour compléter ces deux approches, nous mettons en place une couche de *dropout* (décrochage) pour simuler l'absence d'information. Cette technique permet de mettre à 0 certains neurones aléatoirement pendant l'apprentissage et évite un sur-apprentissage. Cette méthode est très utilisée dans les réseaux multi-dimensionnels comme (Pham et al. 2014).

Les résultats obtenus en bruitant les n-grammes de caractères sont présentés dans la table 5.8. Nous concentrons les expériences sur les apprentissages réalisés avec GCI et GCI+ESP, car ce sont ceux qui ont présenté les meilleurs résultats avec le modèle initial. Afin de faciliter la lecture des résultats en fonction des méthodes utilisées, nous avons coloré les méthodes utilisées pour chaque expérience. Afin de comparer plus facilement ces nouveaux résultats, pour chaque ensemble de validation, les résultats avec le modèle initial (c'est-à-dire toutes les nouvelles fonctions grisées) sont rappelés.

Intuitivement, le fait de bruiter artificiellement le signal des n-grammes de caractères doit dégrader les résultats. En comparant les résultats du modèle initial (lignes entièrement grisées) avec les 4 combinaisons de méthodes employées, nous notons une légère baisse globale avec deux exceptions, ce qui confirme cette intuition. Dans le cadre des expériences menées avec GCI+ESP sur RM et ESP, les meilleurs résultats sont obtenus avec la couche de bruit ajoutée au modèle initial. Cependant, la variation reste peu significative entre les deux meilleurs résultats c'est-à-dire $\pm 0,5\%$.

En comparant les effets entre la fonction de bruit que nous ajoutons au signal d'entrée et la couche gaussienne, il apparaît clairement que les résultats sont beaucoup plus dégradés en utilisant la fonction de bruit. Lorsqu'elle est utilisée sans le décrochage, cas où elle perd le mois de point, elle perd en moyenne 21,53 % sur le taux de caractères correctement reconnus. Le taux de caractères reconnus ne descend pas sous la limite de 65 %. Quant au taux de mots reconnus sans dictionnaire, il chute sous les 15 %. Ces valeurs suivent la même dynamique que les moins bons apprentissages que nous avons expérimentés précédemment (voir table 5.6). L'utilisation du décrochage fait perdre en moyenne 10 % sur le TRC, ce qui induit une grande perte sur le TRM sans dictionnaire avec une perte de 30 % en moyenne.

En comparant les résultats pour chaque ensemble de validation, il apparaît que l'utilisation de ces méthodes n'a pas bouleversé l'apprentissage réalisé. Le meilleur ensemble d'apprentissage pour GCI, RM et ESP reste GCI+ESP.

Au-delà de l'apprentissage par transfert de connaissances, le but de ces méthodes

Apprentissage	Val	Expe. Id	Fonct. Gaussiennne	Couche Gaussiennne	Décrochage	Validation		
						TRC	TRM	TRM dict.
GCI+ESP			False	False	False	**96,49**	**81,45**	**67,22**
	RCI	DB1	True	False	False	72,37	10,44	18,51
		DB2	False	True	False	95,93	78,47	66,1
		DB3	False	True	True	86,2	40	45,86
		DB4	True	False	True	71,21	9,52	19,69
			False	False	False	**93,89**	74,33	36,05
	RM	DB5	True	False	False	72,13	12,42	10,57
		DB6	False	True	False	93,79	**75,09**	**36,94**
		DB7	False	True	True	86,96	47,71	28,41
		DB8	True	False	True	69,43	8,98	10,32
			False	False	False	98,25	91,61	62,42
	ESP	DB9	True	False	False	73,16	10,97	15,81
		DB10	False	True	False	**98,51**	**92,26**	**62,58**
		DB11	False	True	True	91,39	58,87	47,74
		DB12	True	False	True	72,26	13,06	17,1
GCI			False	False	False	**95,85**	**77,83**	**66,47**
	RCI	DB13	True	False	False	75,49	15,63	24,34
		DB14	False	True	False	95,68	76,68	65,66
		DB15	False	True	True	86,58	41,53	45,62
		DB16	True	False	True	67,66	6,68	14,95
			False	False	False	**93,93**	**75,48**	**36,56**
	RM	DB17	True	False	False	75,84	18,47	14,84
		DB18	False	True	False	93,46	73,38	36,18
		DB19	False	True	True	86,77	47,39	27,2
		DB20	True	False	True	69,23	5,67	11,4
			False	False	False	**88,18**	**54,35**	**45,48**
	ESP	DB21	True	False	False	68,34	6,93	8,22
		DB22	False	True	False	87,55	52,42	42,1
		DB23	False	True	True	78,94	28,55	27,74
		DB24	True	False	True	65,87	7,74	8,55

TABLE 5.8 – Résultats pour la génération de séquence expérimentant l'apprentissage par transfert de connaissances avec un signal d'entrée bruité suivant deux méthodes différentes et un décrochage. Les meilleurs résultats pour une ressource d'apprentissage et de validation sont en gras, et les seconds meilleurs résultats sont surlignés.

était de dégrader le signal d'entrée du décodeur afin de se rapprocher au mieux du signal sortant du modèle optique. Nous pouvons faire à ce stade l'hypothèse suivante : bien que l'utilisation de la fonction gaussienne dégrade plus grandement les résultats, c'est peut-être la méthode qui se rapproche le plus de la sortie de l'encodeur. La variation de caractères et de mots bien reconnus avec la couche gaussienne laisse à penser que le signal reste trop proche de l'entrée "idéale" que nous fournissons initialement. Nous vérifierons cette théorie ainsi que les réels effets de ces méthodes, dans la section 5.5, où les deux modèles seront alignés.

Réseau Bidirectionnel Nous avons écarté dès le départ la possibilité d'utiliser une architecture bidirectionnelle pour la partie récurrente du décodeur. Nous avions supposé que toutes les informations contenues dans le vecteur étaient suffisantes à la génération de la séquence. Nous réalisons une expérience pour confirmer cette théorie en ajoutant une couche récurrente GRU *backward*, qui est ensuite concaténée avec la couche *forward* vers la couche *Softmax*. L'expérience est menée sur les ensembles de validation ESP et de test RCI avec l'ensemble d'apprentissage GCI uniquement. Les résultats sont rapportés dans la table 5.9.

Test	**Expe. Id**	**TRC**	**TRM**	**TRM dict.**
RCI	DCI2	**97.10**	**86.22**	**39.30**
	BDD1	96,02	81,89	38,81
ESP	DCI18	88.18	54.35	**45.48**
	BDD2	**88,72**	**55,24**	38,25

TABLE 5.9 – Résultats pour un réseau bidirectionnel GRU avec l'ensemble d'apprentissage GCI identifié par BDD_i. Les résultats obtenus précédemment sont référencés par leur identifiant.

Sur l'ensemble RCI, l'utilisation du réseau bidirectionnel n'apporte aucune amélioration par rapport au réseau initial avec une unique couche *forward*. Sur l'ensemble ESP, les taux de caractères et mots correctement reconnus sans dictionnaire augmentent très légèrement. Le peu de variation que nous constatons ici avec l'utilisation d'un réseau bidirectionnel conforte notre hypothèse de départ selon laquelle l'identification des n-grammes de caractères dans le vecteur suffit à générer le mot.

Plongement de n-grammes de caractères Pour réaliser le plongement des 50 000 n-grammes de caractères, les auteurs de (BENGIO et HEIGOLD 2014) ont défini un réseau composé de 3 couches entièrement connectées de 1 024 neurones utilisant la

fonction d'activation ReLU. Dans notre étude, nous nous sommes limités à 12 500 n-grammes donc nous avons estimé que 500 neurones par couche pouvaient être suffisants. Cependant, nous souhaitons vérifier si le nombre de couches pour opérer le plongement est suffisant avec une unique couche ou si compacter les informations un peu plus pourrait aider à améliorer les résultats. Pour cela, nous avons réalisé une nouvelle série d'expériences sur les ensembles de validation ESP et GCI dont les résultats sont présentés dans la table 5.10. Nous avons utilisé les différentes architectures vues précédemment avec la fonction gaussienne (FG), la couche gaussienne (CG) ainsi que l'architecture initiale, auxquelles nous avons ajouté une couche supplémentaire de la même taille c'est-à-dire 500 neurones. Finalement, l'ajout d'une seconde couche de plongement ne permet pas d'améliorer les résultats obtenus précédemment, et cela sur les différentes architectures expérimentales réalisées. La structure simple définie initialement suffit à notre modèle.

Test	**Apprentissage**	**Expe. Id**	**Nb plongement**	**Couche Gaussienne**	**TRC**	**TRM**	**TRM dict.**
GCI	GCI		1	×	95,85	77,83	66,47
		DE1	2	×	95,82	78,31	65,66
		DE2	2	CG	95,64	77,25	65,32
		DE3	2	FG	59,27	0	2,3
	GCI+ESP		1	×	96,49	81,46	67,22
		DE4	2	×	96,18	79,97	66,75
		DE5	2	CG	96,29	80,74	66,68
		DE6	2	FG	58,95	0	1,33
ESP	GCI		1	×	88,18	54,35	45,48
		DE7	2	×	87,56	51,77	43,22
		DE8	2	CG	88,78	58,06	43,55
		DE9	2	FG	69,18	9,37	13,02
	GCI+ESP		1	×	98,25	91,61	62,42
		DE10	2	×	98,28	92,74	62,42
		DE11	2	CG	98,59	93,71	62,58
		DE12	2	FG	72,31	12,86	14,13

TABLE 5.10 – Résultats avec une modification de la structure sur la couche de plongement pour les ensembles de validation GCI et ESP

5.4.5 Conclusion

Dans cette section, nous avons présenté l'architecture initialement mise en place pour le générateur de séquence, ainsi que différentes variantes visant à améliorer les performances, le tout en appliquant un apprentissage par transfert

de connaissances. Le modèle prend comme entrée une version de la séquence déconstruite au niveau n-grammes de caractères qui doit être assez robuste pour générer la séquence et absorber les mots hors-vocabulaire dû à l'apprentissage par transfert de connaissances.

Nos résultats montrent que l'approche est opérationnelle au niveau mot. Nous obtenons ainsi des TRC supérieurs à 90 % et des TRM dépassant la couverture lexicale estimée. Le décodeur initial est simple de par sa construction avec uniquement 4 couches, mais nous obtenons de bons résultats. Malgré nos tentatives pour enrichir cette architecture avec un réseau bidirectionnel et plus de couches de plongement, le réseau initial a fourni les meilleurs résultats.

Cette section a permis l'évaluation de l'apprentissage par transfert de connaissances en utilisant différentes ressources sous-exploitées et/ou disponibles. Les résultats montrent que réaliser un apprentissage avec une ressource historique pour l'appliquer sur une ressource moderne est possible. Il en est de même pour le passage d'une langue à une autre. Cela renforce l'idée de rechercher de nouvelles ressources inexploitées au lieu de s'appuyer sur des ressources traditionnellement utilisées telles que Wikipédia.

La section suivante présente les expérimentations reliant les deux modèles encodeur et décodeur. Les effets des différents types d'architectures et méthodes utilisés pour l'apprentissage du générateur de séquence y sont mesurés avec la sortie réelle du modèle optique.

5.5 Modèle encodeur-décodeur

Dans les expériences précédentes, nous avons pu observer que le modèle optique fournissait de meilleurs résultats (rappel et F-mesure) avec l'utilisation des unigrammes uniquement tandis que le générateur de séquence a besoin de tous les 1,2,3-grammes de caractères pour obtenir des taux de reconnaissance de caractères et de mots supérieurs à 90 % et 70 %, respectivement. La dernière étape consiste à tester le modèle encodeur-décodeur dans son intégralité. Pour cela, nous sélectionnons et combinons toutes les meilleures configurations pour chaque composant.

5.5.1 Évaluation de l'impact des n-grammes de caractères et des ressources d'apprentissage

La table 5.11 présente les dernières expériences. Nous sélectionnons différentes expériences avec les unigrammes comme pivot (E0-DB17), car l'encodeur a obtenu le meilleur score de F-mesure, les 1,2-grammes avec et sans transfert de connaissances (E1-DB18, B2-DB18), car c'est le meilleur compromis entre les deux composants et les 1,2,3-grammes combinant différentes architectures d'encodeur (E2-DB19, E3-DB19, E6-DB19), car elles ont obtenu le meilleur résultat sur le décodeur.

À première vue, nous trouvons que les résultats obtenus sur la partie encodeur sont similaires à ceux obtenus lors de l'entraînement. Pour aider le système et observer son comportement, nous avons réalisé une étape d'apprentissage à travers le composant encodeur et décodeur à partir des poids précédents afin d'appliquer une rétro-propagation de bout en bout. Ce sont les résultats présentés dans la table 5.11. Nous avons pu constater que cela améliore les performances de l'encodeur lorsque toutes les images sont utilisées, mais dégrade le décodeur lorsque nous l'avons réalisé avec RM et GW, de sorte que le vocabulaire ESP est oublié.

Les résultats obtenus avec l'encodeur entraîné sur ESP donnentla borne haute que les autres systèmes ne dépassent pas c'est-à-dire 44,72 % de F-mesure et 28 % de caractères bien reconnus avec l'aide du dictionnaire. Il apparaît clairement que la qualité de l'encodeur dégrade le potentiel du générateur de séquence. L'analyse détaillée des prédictions montre que la sortie bruitée de l'encodeur empêche le

Exp.		Encodeur			Décodeur
Enc.	**Dec.**	**Rec.**	**Pre.**	**F1-Sc**	**TRC dict.**
E0	DE17	32,15	29,78	30,83	27,01
E1	DE18	29,75	29,47	29,54	**28,07**
E2	DE19	11,03	32,64	16,43	24,39
E4	DE19	3,42	40,22	6,27	25,70
E6	DE19	9,71	33,87	15,02	17,23
E4	DE23	11,02	32,73	16,44	20,24
B2	DE18	**43,61**	45,88	**44,72**	28,02

TABLE 5.11 – Résultats de l'apprentissage par transfert de connaissances sur l'ensemble de test ESP avec les deux modèles placés bout à bout : les poids sauvegardés des expériences précédentes sont utilisés pour initialiser le système complet.

décodeur de bien fonctionner. Par exemple, l'encodeur prédit plusieurs symboles de début et fin du mot ou aucun symbole de fin. De plus, à l'exception des mots courants tels que "de", l'encodeur prédit 7 fois plus de n-grammes de caractères, ce qui rend plus difficile la reconnaissance des mots.

5.5.2 Analyse de l'impact de l'apprentissage bruité du décodeur

La sortie bruitée de l'encodeur est un problème que nous avions pressenti comme un frein au bon fonctionnement du décodage. La table 5.12 présente les résultats des expériences opérées avec les modèles bruités du décodeur tout en faisant varier le seuil d'acceptation des n-grammes de caractères, entre la valeur de décalage 0,5 et le seuil idéal mesuré de 0,2. Les résultats sur ESP, sans utiliser de *fine-tuning*, en utilisant les modèles dont l'apprentissage a été bruité, sont semblables à ceux de la table 5.11. De plus, le changement du seuil d'évaluation n'a aucun effet notable. Finalement, nous constatons que, le *fine-tuning* sur le décodeur uniquement (symbolisé par le symbole suivant " ✓"), peut être une étape indispensable puisque grâce à cette opération, nous observons une augmentation du TRC supérieur à 50 %. La difficulté avec cette méthode est le choix de la ressource, car si nous utilisons une ressource non adaptée, le décodeur risque de perdre l'apprentissage préalable qui a été fait avec des ressources linguistiques dédiées. Le TRM est proche de 0 : c'est pour cette raison qu'il n'a pas été noté. Une analyse détaillée des résultats nous a montré que les mots courts sont correctement identifiés, mais ils sont trop peu nombreux pour peser dans la mesure. De plus, l'opération de *fine-tuning* aide le modèle à prédire des séquences de caractères de la même longueur que celles attendues même si tous les caractères ne sont pas identifiés.

Configuration			*Fine-tuning*	Décodeur	
Modèle	Type bruit	Seuil		**TRC**	**TRC dict.**
2048 Entropie Offset	Couche gaussienne Dropout	0,5	×	0,27	29,30
			✓	57,41	46,39
		0,2	×	0,70	29,46
			✓	61,23	48,6

TABLE 5.12 – Résultats de l'apprentissage par transfert de connaissances sur l'ensemble de test ESP avec l'initialisation du modèle avec les poids de l'expérimentation DB11 (couche gaussienne et décrochage) pour le modèle décodeur et E6 pour l'encodeur avec un seuil de 0,5 pour la validation du vecteur, puis avec un seuil de 0,2.

Finalement, la sortie bruitée de l'encodeur est un réel problème pour le décodeur. C'est le risque avec ce type de structure, la liaison entre les deux composants étant un point sensible et déterminant dans la réussite ce modèle.

5.6 Conclusion

Dans ce chapitre, nous avons proposé un nouveau modèle type encodeur-décodeur avec l'apprentissage par transfert de connaissances. Ce modèle, inspiré de ce qui existe dans l'état-de-l'art dans d'autres domaines, permet un apprentissage indépendant des deux composants. Son point fort est le vecteur pivot qui permet de projeter les informations dans un espace représentant les n-grammes de caractères. Les n-grammes de caractères sont sélectionnés en se basant sur les ressources en notre possession et ont une taille maximum de 3. Ils intègrent également les informations de début et fin de mot. Leur utilisation a déjà fait ses preuves dans d'autres domaines et ils ont prouvé leur robustesse pour les mots hors-vocabulaire. Cela est un grand avantage dans notre étude d'apprentissage par transfert de connaissances multilingue et multi-époque.

Le premier composant que nous avons réalisé est le modèle optique. Il est constitué d'une partie de RNC mettant en œuvre 7 convolutions afin d'extraire les caractéristiques des images de mots passées en entrée du système. Puis, deux solutions ont été envisagées pour vectoriser les caractéristiques. La première méthode est une couche de sous-échantillonnage pyramidale spatiale. Cependant, le découpage opéré n'est pas optimal lorsque les mots ne sont pas centrés dans l'entrée du système. C'est pourquoi nous avons mis en place une seconde solution à base de masques prenant en compte la position des mots dans l'entrée du modèle ainsi que des informations supplémentaires comme les notions de début, milieu et fin de mot. Ces masques représentent un modèle d'attention artificiel qui peut être appris par un réseau à convolution-déconvolution pour segmenter les zones. Le réseau se termine par deux couches entièrement connectées avant de construire le vecteur de n-grammes de caractères. Les expérimentations du composant ont permis de montrer que les meilleures configurations pour un apprentissage classique ne sont pas les mêmes pour des modèles devant réaliser un apprentissage par transfert de connaissances.

Le second composant génère une séquence de caractères à partir du vecteur de n-grammes de caractères passé en entrée. Il est composé d'une couche entièrement connectée pour faire un plongement des n-grammes, mais sans passer par un modèle pré-entrainé. Puis, la couche récurrente suivie d'un *softmax* permet de redonner de la temporalité en générant une succession de caractères jusqu'à atteindre le

symbole de fin de séquence. Ce modèle indépendant de l'autre permet d'utiliser des ressources linguistiques sous-exploitées dans le domaine de l'étude. Les expériences ont montré de bons TRC et TRM, sans dictionnaire, dans le cadre de l'apprentissage par transfert de connaissances. Même en bruitant artificiellement les données afin de se rapprocher au mieux de la sortie réelle de l'encodeur, les TRC restent au-dessus de 65 %. En revanche, les TRM varient énormément de 10 % à 90 % selon le style du bruit qui est mis en place.

Finalement, nous avons combiné les deux composants entraînés pour opérer l'apprentissage par transfert de connaissances. Les premiers résultats révèlent la réelle difficulté de cette combinaison. En effet, les TRC et TRM chutent complètement lorsque les modèles du décodeur utilisant les entrées idéales des vecteurs de n-grammes sont utilisés. De plus, les résultats en mode apprentissage par transfert de connaissances des différents modèles optiques n'étaient pas très élevés, ce qui accentue ces mauvais résultats. La sensibilisation du décodeur aux réelles valeurs de la sortie du modèle optique avec le *fine-tuning* permet d'atteindre de nouveaux TRC acceptables. Cette approche encodeur-décodeur complémentaire, mais indépendante est intéressante pour réaliser un apprentissage par transfert de connaissances. En effet, il permet d'utiliser des ressources disponibles ou de nouvelles ressources pour les apprentissages en mélangeant les langues et les époques. Cependant, il reste des améliorations à apporter principalement à la partie modèle optique, pour construire un vecteur de n-grammes de caractères plus précis.

6

Conclusion et perspectives

Dans ce document, nous avons présenté les travaux relatifs à la création d'un système de reconnaissance d'écriture manuscrite tentant d'exploiter aux mieux les dernières méthodes mises en œuvre en traitement du langage pour des documents historiques et multilingues dans le domaine de la Comédie-Italienne. Contrairement aux approches classiques de l'état de l'art en reconnaissance d'écriture manuscrite, nous avons souhaité ne pas utiliser simplement le modèle linguistique pour contraindre la sortie du modèle optique, mais bien d'exploiter au maximum l'ensemble des capacités de chacun des systèmes. De plus, ce système doit répondre à une forte contrainte qui est l'absence (totale ou presque) de données annotées sur l'étude réalisée.

6.1 Synthèse

Une partie de notre étude s'est concentrée sur la constitution d'une base de test de titres de la Comédie-Italienne réalisée grâce à la plateforme participative RECITAL. Nous y avons récupéré les informations de localisations et de transcriptions candidates relatives aux zones de titres indiquées par les annotateurs. Nous avons ensuite converti ces informations au format PiFF (Pivot File Format) afin de faciliter leur exploitation. Pour pouvoir se placer dans un contexte d'images de lignes, nous avons appliqué un algorithme de segmentation automatique en lignes

de chaque zone identifiée par les utilisateurs. Finalement, nous avons sélectionné et vérifié 971 images lignes de titres annotées couvrant un maximum de registres. Cependant, cette quantité d'images reste insuffisante pour pouvoir réaliser un apprentissage adéquat d'un réseau de neurones profonds tel que ceux que nous avons présentés dans le chapitre 2.

Forts de ce constat, nous avons orienté nos travaux vers l'apprentissage par transfert de connaissances qui semble être la solution lorsque l'on doit faire face à un manque (voire une absence) de données pour l'apprentissage d'un modèle. Dans notre cas précisément, nous devons réaliser un transfert transductif qui consiste à entraîner un système pour une tâche précise à partir de données sources, et à appliquer le modèle sur des données ayant une provenance différente, mais ayant un lien avec les données d'apprentissage utilisées. Dans ce but, nous avons sélectionné trois bases de données annotées et disponibles, et ayant au moins un point en commun avec nos données : la langue, l'époque ou la qualité des images.

Dans le chapitre 4, nous avons tout d'abord expérimenté un premier système constitué d'un réseau à convolutions pour extraire automatiquement les caractéristiques des images de mots présentées en entrée du système, suivi par une partie BLSTM-CTC qui permet de générer une séquence de caractères correspondant au mot contenu dans l'image. Plusieurs configurations ont été expérimentées afin de trouver celle favorisant l'apprentissage par transfert de connaissances. Finalement, la meilleure configuration obtenue, lors d'un apprentissage classique, mais mélangeant les données multilingues, nous a permis de reconnaître correctement seulement 10,6% de caractères dans les lignes de titres de la Comédie-Italienne ce qui n'est pas suffisant pour reconnaître des mots. Pour tenter d'améliorer ce score, mais également d'évaluer la force de notre architecture sur les images de lignes, nous avons initialisé les poids du système appris sur RIMES et Les Esposalles, et nous avons poursuivi l'entraînement sur une faible quantité d'images de la Comédie-Italienne. Cela a permis d'améliorer de 18% le taux de reconnaissance des caractères. L'analyse des erreurs nous a montré que seuls les mots proches dans les langues sources et de tests étaient bien reconnus. Cela nous a conduits à émettre des suppositions sur le rôle implicite que jouent les cellules LSTM dans l'apprentissage d'un modèle de langage.

Dans le but d'atteindre notre principal objectif, c'est-à-dire allier les compétences des dernières méthodes en traitement du langage et en reconnaissance d'écriture manuscrite, nous avons proposé une nouvelle solution séparant explicitement en deux composantes le système présenté dans le chapitre 5. Notre principal apport est la définition de ce modèle permettant l'apprentissage par transfert de connaissances. Comme cela est fait dans d'autres domaines d'applications, nous avons défini un système de type encodeur-décodeur : un modèle optique qui se concentre sur

les caractéristiques de l'image, et un modèle linguistique générant des séquences de caractères. L'originalité de cette approche est la projection des informations dans un espace non-latent commun aux deux parties du système favorisant ainsi un apprentissage indépendant. Cet espace commun, qui est le pivot du système, représente les n-grammes de caractères d'une longueur maximum de 3 et indique les début et fin de mot. L'utilisation des n-grammes de caractères a été motivée par les performances qu'ils ont réalisées dans d'autres domaines en traitement du langage pour gérer les grandes quantités de mots hors-vocabulaire.

Pour la modélisation du système, nous avons choisi de conserver un réseau à convolutions pour extraire automatiquement les caractéristiques des images. Cela permet d'éviter de pré-traiter les images utilisées pour l'apprentissage et ainsi conserver leurs différences qui sont un avantage pour généraliser au mieux le modèle. Des masques sont ensuite utilisés pour cibler des zones dans les caractéristiques extraites et encoder les informations dans un vecteur de n-grammes de caractères, indiquant ainsi les n-grammes composant le mot de l'image. Les différents apprentissages et évaluations du composant nous ont permis de mettre en évidence qu'un modèle qui montre de bons résultats pour un apprentissage classique, c'est-à-dire avec des données provenant de la même ressource pour l'apprentissage et le test, présente des difficultés à opérer l'apprentissage par transfert de connaissances. De plus, la longueur du vecteur de n-grammes de caractères joue un rôle déterminant dans les résultats. En effet, les expériences ont montré que restreindre la sortie au seul objectif des unigrammes était plus performant qu'en considérant l'ensemble des n-grammes de caractères.

Le décodeur, quant à lui, prend ce vecteur de n-grammes de caractères pour générer une séquence de caractères correspondant au mot de l'image. Pour l'apprentissage, nous avons cherché des ressources supplémentaires et encore sous-exploitées traitant de la Comédie-Italienne du XVIII^{ème} siècle. À partir de 23 œuvres sélectionnées sur Google Livre, nous avons créé une nouvelle ressource linguistique contenant plus de 25 000 entrées avec une orthographe de l'époque, comportant beaucoup d'entités nommées comme des noms de pièces, des acteurs ou des personnages. Nous avons testé les performances du décodeur dans différentes conditions d'apprentissage en mélangeant les langues, les époques des ressources, en faisant varier la longueur des n-grammes de caractères utilisée ainsi qu'en bruitant l'entrée du modèle pour se rapprocher de la véritable sortie de l'encodeur. Sur les transcriptions des images des registres de la Comédie-Italienne , le modèle de base utilisant tous les n-grammes de caractères a permis de reconnaître 97,27 % des caractères et 86,57 % de mots sans dictionnaire. Cependant suivant les méthodes pour bruiter le signal d'entrée, les résultats sont dégradés avec une chute du TRC à 65 % et un TRM variant très largement entre 10 et 90 %.

Les deux modèles composant notre système complet de reconnaissance d'écriture manuscrite ont chacun montré de bonnes performances, mais dans des conditions différentes. En effet, les deux réseaux se rejoignent sur le vecteur de n-grammes de caractères or dans un cas les résultats sont meilleurs avec seulement les unigrammes tandis que dans l'autre, l'ensemble des n-grammes est requis pour pouvoir reconstruire et reconnaître un maximum de mots. Une fois que ces deux composants sont reliés, les résultats montrent les points faibles de notre idée pour appliquer apprentissage par transfert de connaissances. En effet, le TRC final avec et sans *fine-tuning* reste très faible, et l'utilisation du décodeur bruité ne permet pas non plus d'atteindre de résultats satisfaisants. Nous avons ciblé plusieurs facteurs pouvant expliquer ces résultats, dont la quantité des n-grammes de caractères sélectionnés, l'architecture de l'encodeur ou encore la qualité de l'entrée du décodeur.

6.2 De l'Apprentissage à la Connaissance

Les perspectives pour obtenir un modèle robuste à l'apprentissage par transfert de connaissances sont nombreuses. Pour commencer, la projection dans un espace non-latent commun aux deux composants du système donne l'avantage de pouvoir multiplier les ressources disponibles. Cependant, il peut être intéressant de lui apporter des modifications afin qu'il soit plus réduit, mais contienne assez d'informations pour couvrir un vocabulaire large et multilingue.

Dans la partie modèle optique, les améliorations nécessaires devront se concentrer sur la transformation des caractéristiques extraites par le réseau à convolution pour venir constituer le vecteur de n-grammes de caractères. Pour remplacer les masques artificiels utilisés permettant de cibler certaines zones de caractéristiques, un générateur de masque du type Convolution-Déconvolution peut être mis en place. Une amélioration pourrait viser à décomposer la projection dans l'espace commun, c'est-à-dire commencer par projeter dans un premier vecteur représentant uniquement les unigrammes puis projeter ces informations vers un nouveau vecteur représentant les bigrammes... Pour finir, pour pouvoir utiliser l'encodeur sur des images de lignes, la plus grande avancée serait d'ajouter un mécanisme d'attention qui ciblerait les mots un à un parcourant ainsi l'ensemble des lignes. Cette méthode fortement utilisée dernièrement permet de s'affranchir de l'étape de segmentation.

Dans la partie décodeur, les modèles de *word embedding* existants en monolingue (et à venir en multilingue) pour les n-grammes de caractères pourraient être adaptés pour venir renforcer le générateur de séquences. Cependant, le problème lié à l'entrée du décodeur ne doit pas être négligé, en effet dans cette partie du système, il est le point déterminant pour favoriser la réussite de la reconnaissance d'écriture

manuscrite. Une analyse de la sortie de l'encodeur pourrait fournir une indication sur la forme du bruit généré, pour ensuite le reproduire sur l'entrée du décodeur.

Cette thèse s'est inscrite volontairement dans une démarche exploratoire. En effet, à la place de suivre la voie classiquement empruntée en reconnaissance d'écriture manuscrite, nous avons profité du fait de débuter de zéro dans ce projet pour nous orienter vers une nouvelle solution. Nos plus grandes contraintes ont été l'absence de connaissances sur les documents étudiés avec l'ensemble des spécificités qui les caractérisent ainsi que notre volonté à se positionner au croisement des domaines du TALN et de la reconnaissance d'écriture pour l'analyse automatique de documents anciens. C'est pour cette raison que ce travail reste original tant dans l'idée que dans la réalisation. Il reste encore beaucoup de pistes à explorer pour trouver un système capable de réaliser un apprentissage par transfert de connaissances performant. Cependant parvenir à définir un modèle assez robuste pour pouvoir prendre des ressources de langues et de domaines différents reste une idée intéressante à développer. Ces dernières années, l'apprentissage par transfert de connaissances est de plus en plus abordé pour pouvoir profiter de l'avantage des réseaux de neurones profonds déjà entrainés et disponibles. Nous espérons que ces travaux ouvriront la voie à d'autres afin d'atteindre des modèles assez généraux et permettre l'exploration et la diffusion du savoir au plus grand nombre. Finalement, les agents intelligents n'ont pas pour seule vocation d'extraire au mieux la connaissance, mais également de la présenter et la rendre accessible à tous. Là où il nous faudrait une vie pour trouver, déchiffrer et comprendre le passé à travers tous les récits qui restent, l'intelligence artificielle apportera un gain de temps significatif dans l'appropriation de la connaissance par tous.

Table des figures

Bibliographie

Abiteboul, Serge, R. Hull et Victor Vianu (1995). *Foundations of Databases*. 685 p. Addison-Wesley. isbn : 0-201-53771-0.

Agius, Harry W. et Marios C. Angelides (sept. 2001). « Spatial Color Indexing Using Rotation, Translation, and Scale Invariant Anglograms ». In : *Multimedia Tools and Applications* 15.1, p. 5-37.

Allen, J. F. (nov. 1983). « Maintaining Knowledge About Temporal Intervals ». In : *Communications of the ACM* 26.11, p. 832-843.

Almazán, Jon et al. (2014a). « Segmentation-free word spotting with exemplar SVMs ». In : *Pattern Recognition* 47.12, p. 3967-3978.

– (2014b). « Word spotting and recognition with embedded attributes ». In : *IEEE transactions on pattern analysis and machine intelligence* 36.12, p. 2552-2566 (cf. p. 94).

– (2014c). « Word spotting and recognition with embedded attributes ». In : *IEEE tran. on PAMI* 36.12, p. 2552-2566.

Álvaro, Francisco et al. (2015). « Structure detection and segmentation of documents using 2D stochastic context-free grammars ». In : *Neurocomputing* 150, p. 147-154.

Alvaro, Francisco et al. (2015). « Structure detection and segmentation of documents using 2D stochastic context-free grammars ». In : *Neurocomputing* 150, p. 147-154. url : http://www.sciencedirect.com/science/article/pii/S0925231214012648.

Antonacopoulos, A et al. (2015). « ICDAR2015 competition on recognition of documents with complex layouts-RDCL2015 ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 1151-1155.

Aradillas, José Carlos, Juan José Murillo-Fuentes et Pablo M Olmos (2018). « Boosting Handwriting Text Recognition in Small Databases with Transfer Learning ». In : *arXiv preprint arXiv :1804.01527*.

Arvanitopoulos, Nikolaos et Sabine Süsstrunk (2014). « Seam carving for text line extraction on color and grayscale historical manuscripts ». In : *ICFHR*, p. 726-731 (cf. p. 57, 60).

Arwa, AL-Khatatneh, Sakinah Ali Pitchay et Musab Al-qudah. « A Review of Skew Detection Techniques for Document ». In : ().

Asi, Abedelkadir, Raid Saabni et Jihad El-Sana (2011). « Text line segmentation for gray scale historical document images ». In : *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. ACM, p. 120-126.

Augustin, Emmanuel et al. (2006). « RIMES evaluation campaign for handwritten mail processing ». In : *ICFHR*.

Badrinarayanan, Vijay, Alex Kendall et Roberto Cipolla (2015). « Segnet : A deep convolutional encoder-decoder architecture for image segmentation ». In : *arXiv preprint arXiv :1511.00561* (cf. p. 105).

Baeza-Yates, R. A. (sept. 1992). « Text retrieval : Theory and practice ». In : *Proceedings of the 12th IFIP World Computer Congress*. Sous la dir. de J. van Leeuwen. T. I. Madrid, Spain : Elsevier Science, p. 465-476. url : citeseer.nj.nec.com/baeza-yates92text.html.

Baeza-Yates, Ricardo, Eduardo F. Barbosa et Nivio Ziviani (sept. 1996). « Hierarchies of indices for text searching ». In : *Information Systems* 21.6, p. 497-514.

Bahdanau, Dzmitry, KyungHyun Cho et Yoshua Bengio (2014). « NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE ». In : (cf. p. 92).

Bal, H. E., J. G. Steiner et A. S. Tanenbaum (sept. 1989). « Programming Languages for Distributed Computing Systems ». In : *ACM Computing Surveys* 21.3, p. 261-322.

Ballard, D. H. et C. M. Brown (1982). *Computer Vision*. 523 p. Prentice-Hall.

Balter, Roland, Jean-Pierre Banâtre et Serge Krakowiak (1994). *Construction des systèmes d'exploitation répartis*. Collection Didactique. 404 p. Institut National de la Recherche en Informatique et Automatique (INRIA).

Baral, Chitta, Graciela Gonzalez et Tran Son (juil. 1998). « Conceptual Modeling and Querying in Multimedia Databases ». In : *Multimedia Tools and Applications* 7.1-2, p. 37-66.

Batini, C., Stefano Ceri et S. B. Navathe (1992). *Conceptual Database Design : An Entity-Relationship Approach*. 470 p. Benjamin/Cummings.

Bayley, P. et D. Hawking (1996). *A Parallel architecture for query processing over a Terabyte of text*. Rapp. tech. The Australian National University, p. 1243-1259.

Beigbeder, Michel et Annabelle Mercier (juin 2003). « Étude des distributions de tf et de idf sur une collection de 5 millions de pages html ». In : *Atelier de recherche d'information sur le passage á l'échelle, congrès INFORSID*. Nancy, France.

Bell, D. et J. Grimson (1992). *Distributed Database Systems*. 410 p. Addison-Wesley.

Bellman, Richard (1961). *Adaptive Control Processes : A Guided Tour*. Princeton University Press.

Bengio, Samy et Georg Heigold (2014). « Word embeddings for speech recognition ». In : *Proceedings of the 15th Annual Conference of the International Speech Communication Association (Interspeech'14)*. Singapore (cf. p. 93, 95, 122).

Bengio, Yoshua et al. (2003). « A neural probabilistic language model ». In : *Journal of machine learning research* 3.Feb, p. 1137-1155 (cf. p. 40).

Berchtold, S., D. A. Keim et H.-P. Kriegel (1996). « The X-tree : An Index Structure for High-Dimensional Data ». In : *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96)*. Mumbai (Bombay), India, p. 28-39.

Bernstein, P. A., V. Hadzilacos et N. Goodman (1987). *Concurrency Control and Recovery in Database Systems*. 685 p. Addison-Wesley. url : http://research.microsoft.com/pubs/ccontrol/.

Berry, Michael W., Z. Darmac et E. R. Jessup (1999). « Matrices, vector spaces and information retrieval ». In : *SIAM* 41.2, p. 335-362.

Bertino, Elisa et al. (1997). *Indexing Techniques for Advanced Database Systems*. 250 p. Boston, Massachussets : Kluwer Academic.

Bertolami, Roman et al. (2007). « Non-uniform slant correction for handwritten text line recognition ». In : *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. T. 1. IEEE, p. 18-22 (cf. p. 25).

Bhaskaran, V. et K. Konstantinides (1995). *Image and Video Compression Standards : Algorithms and Architectures*. 369 p. Prentice-Hall.

Bideault, Gautier et al. (2015). « Benchmarking discriminative approaches for word spotting in handwritten documents ». In : *ICDAR*, p. 201-205.

Blok, H.E., A.P. de Vries et H.M. Blanken. *Top N MM query optimization - The best of both IR and DB worlds*. url : citeseer.ist.psu.edu/blok99top.html.

Bluche, Théodore (2015). « Deep Neural Networks for Large Vocabulary Handwritten Text Recognition ». Thèse de doct. Université Paris Sud - Paris XI.

Bluche, Théodore et Ronaldo Messina (2017). « Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 646-651 (cf. p. 27).

Bluche, Théodore, Hermann Ney et Christopher Kermorvant (2013). « Tandem HMM with convolutional neural network for handwritten word recognition ». In : *ICASSP*, p. 2390-2394.

Bluche, Théodore et al. (2015). « Framewise and CTC training of Neural Networks for handwriting recognition ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 81-85 (cf. p. 36).

Bluche, Théodore et al. (2017a). « Cortical-Inspired Open-Bigram Representation for Handwritten Word Recognition ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 73-78 (cf. p. 95).

Bluche, Théodore, Jérôme Louradour et Ronaldo Messina (2017b). « Scan, Attend and Read : End-to-End Handwritten Paragraph Recognition with MDLSTM Attention ». In : *Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR'17)*. Kyoto, Japan (cf. p. 92).

Bobrowski, S. (1998). *ORACLE 8 Architecture : Understand, Plan, and Migrate to Oracle's Revolutionary Database*. 356 p. Osborne McGraw-Hill & Oracle Press.

Bojanowski, Piotr et al. (2016). « Enriching word vectors with subword information ». In : *arXiv preprint arXiv :1607.04606* (cf. p. 40).

– (2017). « Enriching Word Vectors with Subword Information ». In : *Transactions of the Association of Computational Linguistics* 5.1, p. 135-146 (cf. p. 114).

Bollmann, Marcel, Joachim Bingel et Anders Søgaard (2017). « Learning attention for historical text normalization by learning to pronounce ». In : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. Vancouver, Canada, p. 332-344. URL : http://aclweb.org/anthology/P17-1031.

Bouchon-Meunier, B. (oct. 1995). *La logique floue et ses applications*. 257 p. Addison-Wesley.

Boukharouba, Abdelhak (2017). « A new algorithm for skew correction and baseline detection based on the randomized Hough Transform ». In : *Journal of King Saud university-computer and information sciences* 29.1, p. 29-38 (cf. p. 24).

Bouzeghoub, Mokrane, Georges Gardarin et Patrick Valduriez (1997). *Object Technology : Concepts and Methods*. Software Engineering Series. 382 p. International Thomson Computer Press.

Bozinovic, Radmilo M. et Sargur N. Srihari (1989). « Off-line cursive script word recognition ». In : *IEEE Transactions on pattern analysis and machine intelligence* 11.1, p. 68-83 (cf. p. 24).

Brenner, Clarence Dietz (1961). *The Théâtre Italien : its reportory, 1716-1793*. University of California Press.

Brigger, P. (1995). « Morphological Shape Representation Using the Skeleton Decomposition : Application to Image ». No. 1448. Thèse de doct. École Polytechnique Fédérale de Lausanne.

Bruijn, Willem de et Michael S. Lew (2002). « AtomsNet : Multimedia Peer2Peer File Sharing ». In : *CIVR*. Sous la dir. de M. S. Lew, N. Sebe et J. P. Eakins. T. LNCS 2383. Springer-Verlag, p. 138-146.

Bryan, M. (1988). *SGML : An Author's Guide to the Standard Generalized Markup Language*. Addison-Wesley.

Bunke, Horst, Markus Roth et Ernst Günter Schukat-Talamazzini (1995). « Off-line cursive handwriting recognition using hidden Markov models ». In : *PR* 28.9, p. 1399-1413 (cf. p. 28).

Bush, Vannevar (juil. 1945). « As We May Think ». In : *The Atlantic Monthly* 1.176, p. 101-108. url : www.isg.sfu.ca/$\tilde{}$duchier/misc/vbush/vbush-all.shtml.

Caesar, Torsten, Joachim M Gloger et Eberhard Mandler (1993). « Preprocessing and feature extraction for a handwriting recognition system ». In : *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. IEEE, p. 408-411 (cf. p. 25).

Campbell, J. P. (1997). « Speaker Recognition : A Tutorial ». In : *Proceedings of the IEEE* 85.9, p. 1437-1462.

Chan, William et al. (2016). « Listen, attend and spell : A neural network for large vocabulary conversational speech recognition ». In : *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, p. 4960-4964 (cf. p. 92).

Chatfield, K. et al. (2014). « Return of the Devil in the Details : Delving Deep into Convolutional Nets ». In : *British Machine Vision Conference*. arXiv : 1405.3531 [cs].

Chen, Kai et al. (2015a). « Page segmentation of historical document images with convolutional autoencoders ». In : *ICDAR*, p. 1111-1115.

– (2015b). « Page segmentation of historical document images with convolutional autoencoders ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 1011-1015.

Chittilappilly, Anand Inasu, Lei Chen et Sihem Amer-Yahia (2016). « A Survey of General-Purpose Crowdsourcing Techniques ». In : *IEEE Transactions on Knowledge and Data Engineering* 28.9, p. 2246-2266. issn : 1041-4347. doi : 10.1109/TKDE.2016.2555805. url : http://ieeexplore.ieee.org/document/7456302/ (cf. p. 50).

Cho, Kyunghyun et al. (2014a). « Learning phrase representations using RNN encoder-decoder for statistical machine translation ». In : *arXiv preprint arXiv :1406.1078* (cf. p. 33, 92, 112).

Cho, Kyunghyun et al. (2014b). « On the Properties of Neural Machine Translation : Encoder–Decoder Approaches ». In : *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST'14)*. Doha, Qatar, p. 103-111. url : http://www.aclweb.org/anthology/W14-4012 (cf. p. 92).

Chung, Junyoung et al. (2014). « Empirical evaluation of gated recurrent neural networks on sequence modeling ». In : *arXiv preprint arXiv :1412.3555* (cf. p. 33).

Clematide, Simon, Lenz Furrer et Martin Volk (2016). « Crowdsourcing an OCR Gold Standard for a German and French Heritage Corpus ». In : p. 975-982.

DOI : 10.5167/UZH-124786. URL : https://search.datacite.org/works/10.5167/UZH-124786 (cf. p. 56).

CLEVERT, Djork-Arné, Thomas UNTERTHINER et Sepp HOCHREITER (2015). « Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) ». In : *CoRR* abs/1511.07289. URL : http://arxiv.org/abs/1511.07289.

CLOPPET, Florence et al. (2016a). « ICFHR 2016 Competition on the Classification of Medieval Handwritings in Latin Script ». In : *ICFHR*, p. 590-595.

CLOPPET, Florence et al. (2016b). « ICFHR2016 Competition on Classification of Medieval Handwritings in Latin Script ». In : *Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR'16)*. Shenzhen, China, p. 590-595.

COATES, A., H. LEE et A.Y. NG (2011). « An analysis of single-layer networks in unsupervised feature learning ». In : *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Sous la dir. de Geoffrey GORDON, David DUNSON et Miroslav DUDÍK. T. 15. JMLR Workshop and Conference Proceedings. JMLR W&CP, p. 215-223. URL : http://jmlr.csail.mit.edu/proceedings/papers/v15/coates11a.html.

CONNEAU, Alexis et al. (2017). « Supervised Learning of Universal Sentence Representations from Natural Language Inference Data ». In : *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 670-680 (cf. p. 44).

COUASNON, Bertrand (2001). « Dmos : A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems ». In : *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, p. 215-220 (cf. p. 48).

COÜASNON, Bertrand et Jean CAMILLERAPP (2002). « DMOS, une méthode générique de reconnaissance de documents : évaluation sur 60 000 formulaires du XIXe siecle ». In : *Colloque international francophone sur l'écrit et le document*, p. 225-234.

DALAL, Navneet et Bill TRIGGS (2005). « Histograms of oriented gradients for human detection ». In : *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. T. 1. IEEE, p. 886-893 (cf. p. 26).

DE LUCA, Emanuele (2011). *Le Répertoire de la Comédie-Italienne (1716-1762)*. Paris :IRPMF.

DELIGNE, Sabine et Frédéric BIMBOT (1997). « Inference of variable-length linguistic and acoustic units by multigrams1 ». In : *Speech Communication* 23.3, p. 223-241 (cf. p. 38).

DING, Haisong et al. (2017). « A compact CNN-DBLSTM based character model for offline handwriting recognition with Tucker decomposition ». In : *Document*

Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. T. 1. IEEE, p. 507-512 (cf. p. 27).

Donahue, Jeff et al. (2014). « Decaf : A deep convolutional activation feature for generic visual recognition ». In : *International conference on machine learning*, p. 647-655 (cf. p. 43).

Donahue, Jeffrey et al. (2015). « Long-term recurrent convolutional networks for visual recognition and description ». In : *CVPR*, p. 2625-2634.

Du, Bo et al. (2017). « Stacked convolutional denoising auto-encoders for feature representation ». In : *IEEE transactions on cybernetics* 47.4, p. 1017-1027 (cf. p. 28).

Espana-Boquera, Salvador et al. (2011). « Improving offline handwritten text recognition with hybrid HMM/ANN models ». In : *IEEE transactions on pattern analysis and machine intelligence* 33.4, p. 767-779 (cf. p. 24).

Fine, Shai, Yoram Singer et Naftali Tishby (1998). « The hierarchical hidden Markov model : Analysis and applications ». In : *Machine Learning* 32.1, p. 41-62 (cf. p. 28).

Fischer, Andreas et al. (2009a). « Automatic transcription of handwritten medieval documents ». In : *VSMM*, p. 137-142 (cf. p. 29).

– (2009b). « Automatic transcription of handwritten medieval documents ». In : *Virtual Systems and Multimedia, 2009. VSMM'09. 15th International Conference on*. IEEE, p. 137-142 (cf. p. 34).

Fischer, Andreas et al. (2012). « Lexicon-free handwritten word spotting using character HMMs ». In : *PRL* 33.7, p. 934-942 (cf. p. 28, 82).

Freeman, William T et Michal Roth (1995). « Orientation histograms for hand gesture recognition ». In : *International workshop on automatic face and gesture recognition*. T. 12, p. 296-301 (cf. p. 26).

Frinken, Volkmar et al. (2010). « Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents ». In : *ICFHR*, p. 352-357 (cf. p. 44).

Frinken, Volkmar et al. (2012). « A novel word spotting method based on recurrent neural networks ». In : *IEEE Trans. on PAMI* 34.2, p. 211-224 (cf. p. 82).

Frinken, Volkmar, Andreas Fischer et Carlos-D Martínez-Hinarejos (2013). « Handwriting recognition in historical documents using very large vocabularies ». In : *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, p. 67-72.

Gari, Ahmed et al. (2017). « Skew detection and correction based on Hough transform and Harris corners ». In : *Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017 International Conference on*. IEEE, p. 1-4.

Garrette, Dan et Hannah Alpert-Abrams (2016). « An Unsupervised Model of Orthographic Variation for Historical Document Transcription ». In : *Proceedings*

of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HTL'16). San Diego, CA, USA, p. 467-472.

Garrette, Dan et al. (2015). « Unsupervised code-switching for multilingual historical document transcription ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1036-1041.

Garz, Angelika et al. (2012). « Binarization-free text line segmentation for historical documents based on interest point clustering ». In : *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, p. 95-99.

Gatos, Basilios, Ioannis Pratikakis et Stavros J Perantonis (2006). « Adaptive degraded document image binarization ». In : *Pattern recognition* 39.3, p. 317-327 (cf. p. 24).

– (2008). « Efficient binarization of historical and degraded document images ». In : *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*. IEEE, p. 447-454 (cf. p. 24).

Ghorbel, Adam, Jean-Marc Ogier et Nicole Vincent (2015). « A segmentation free Word Spotting for handwritten documents ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 346-350.

Granell, Emilio et al. (2018). « Transcription of Spanish Historical Handwritten Documents with Deep Neural Networks ». In : *Journal of Imaging* 4.1, p. 15 (cf. p. 35).

Granet, Adeline et al. (2018a). « Crowdsourcing-based Annotation of the Accounting Registers of the Italian Comedy ». Anglais. In : (cf. p. 48, 72).

Granet, Adeline et al. (2018b). « Transfer Learning for Handwriting Recognition on Historical Documents ». In : *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods ICPRAM*, p. 432-439 (cf. p. 89).

Granet, Adeline et al. (jan. 2018c). « Transfer Learning for Handwriting Recognition on Historical Documents ». In : URL : https://hal.archives-ouvertes.fr/hal-01681126.

Graves, Alex (2012a). « Sequence transduction with recurrent neural networks ». In : *arXiv preprint arXiv :1211.3711* (cf. p. 36).

– (2012b). *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer (cf. p. 36, 77, 78).

– (2013). « Generating sequences with recurrent neural networks ». In : *arXiv preprint arXiv :1308.0850*.

Graves, Alex et Jürgen Schmidhuber (2005). « Framewise phoneme classification with bidirectional LSTM and other neural network architectures ». In : *Neural Networks* 18.5-6, p. 602-610.

– (2009). « Offline handwriting recognition with multidimensional recurrent neural networks ». In : *NIPS*, p. 545-552 (cf. p. 35, 85).
Graves, Alex, Santiago Fernández et Jürgen Schmidhuber (2005). « Bidirectional LSTM networks for improved phoneme classification and recognition ». In : *ICANN*, p. 799-804.
Graves, Alex et al. (2006). « Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks ». In : *ICML*, p. 369-376 (cf. p. 35).
Graves, Alex, Santiago Fernández et Jürgen Schmidhuber (2007). « Multi-dimensional Recurrent Neural Networks ». In : *Proceedings of the 17th International Conference on Artificial Neural Networks*. ICANN'07. Porto, Portugal : Springer-Verlag, p. 549-558. isbn : 3-540-74689-7, 978-3-540-74689-8. url : http://dl.acm.org/citation.cfm?id=1776814.1776875 (cf. p. 35).
Graves, Alex et al. (2009). « A novel connectionist system for unconstrained handwriting recognition ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.5, p. 855-868 (cf. p. 34, 36, 77, 80).
Grosicki, E. et H. El-Abed (2011). « ICDAR 2011 : French handwriting recognition competition ». In : *ICDAR*, p. 1459-1463 (cf. p. 65).
Grosicki, Emmanuele et Haikal El Abed (2009a). « ICDAR 2009 handwriting recognition competition ». In : *ICDAR*, p. 1398-1402 (cf. p. 29).
Grosicki, Emmanuèle et Haikal El Abed (2009b). « ICDAR 2009 handwriting recognition competition ». In : *ICDAR*, p. 1398-1402 (cf. p. 35, 77).
He, Dafang et al. (2017). « Multi-Scale Multi-Task FCN for Semantic Page Segmentation and Table Detection ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 254-261 (cf. p. 27).
He, Kaiming et al. (2014). « Spatial pyramid pooling in deep convolutional networks for visual recognition ». In : *European conference on computer vision*. Springer, p. 346-361 (cf. p. 99).
– (2016). « Deep residual learning for image recognition ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770-778 (cf. p. 27, 76).
Hochreiter, Sepp et Jürgen Schmidhuber (1997a). « Long short-term memory ». In : *Neural computation* 9.8, p. 1735-1780 (cf. p. 32).
– (1997b). « Long short-term memory ». In : *Neural Computation* 9.8, p. 1735-1780.
Hochreiter, Sepp et al. (2001). « A field guide to dynamical recurrent neural networks ». In : IEEE Press. Chap. Gradient flow in recurrent nets : the difficulty of learning long-term dependencies (cf. p. 32).
Hough, Paul VC (1962). *Method and means for recognizing complex patterns*. Rapp. tech.

Howe, Nicholas R, Shaolei Feng et R Manmatha (2009). « Finding words in alphabet soup : Inference on freeform character recognition for historical scripts ». In : *Pattern Recognition* 42.12, p. 3338-3347 (cf. p. 26).

Hu, Fan et al. (2015a). « Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery ». In : *Remote Sensing* 7.11, p. 14680-14707 (cf. p. 43).

– (2015b). « Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery ». In : *Remote Sensing* 7.11, p. 14680. ISSN : 2072-4292. DOI : 10.3390/rs71114680. URL : http://www.mdpi.com/2072-4292/7/11/14680.

Huang, Po-Sen et al. (2013a). « Learning deep structured semantic models for web search using clickthrough data ». In : *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, p. 2333-2338 (cf. p. 93, 95, 116).

– (2013b). « Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data ». In : *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA : ACM, p. 2333-2338. ISBN : 978-1-4503-2263-8. DOI : 10.1145/2505515.2505665. URL : http://doi.acm.org/10.1145/2505515.2505665 (cf. p. 97).

Irie, Kazuki et al. (2016). « LSTM, GRU, Highway and a Bit of Attention : An Empirical Overview for Language Modeling in Speech Recognition ». In : *INTERSPEECH*. ISCA, p. 3519-3523 (cf. p. 33).

Jia, Yangqing et al. (2014). « Caffe : Convolutional Architecture for Fast Feature Embedding ». In : *Proceedings of the 22Nd ACM International Conference on Multimedia*. MM '14. Orlando, Florida, USA : ACM, p. 675-678. ISBN : 978-1-4503-3063-3. DOI : 10.1145/2647868.2654889. URL : http://doi.acm.org/10.1145/2647868.2654889.

Karpathy, Andrej et Li Fei-Fei (2015). « Deep Visual-Semantic Alignments for Generating Image Descriptions ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. Hawaii, USA, p. 3128-3137.

Karpathy, Andrej et al. (2014). « Large-Scale Video Classification with Convolutional Neural Networks ». In : *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '14. Washington, DC, USA : IEEE Computer Society, p. 1725-1732. ISBN : 978-1-4799-5118-5. DOI : 10.1109/CVPR.2014.223. URL : http://dx.doi.org/10.1109/CVPR.2014.223.

Katz, Slava (1987). « Estimation of probabilities from sparse data for the language model component of a speech recognizer ». In : *IEEE transactions on acoustics, speech, and signal processing* 35.3, p. 400-401 (cf. p. 39).

Kennard, Douglas J et William A Barrett (2006). « Separating lines of text in free-form handwritten historical documents ». In : *Document Image Analysis for Libraries, 2006. DIAL'06. Second International Conference on*. IEEE, 12-pp.

Kessentini, Yousri, Thierry Paquet et AbdelMajid Ben Hamadou (2009). « A multi-lingual recognition system for arabic and latin handwriting ». In : *ICDAR*, p. 1196-1200.

Ketata, Dalel et Maher Khemakhem (2010). « Un survol sur l'analyse et la reconnaissance de documents : imprimé, ancien et manuscrit ». In : *Colloque International Francophone sur l'Ecrit et le Document (CIFED2010)*, 12pages (cf. p. 24).

Koerich, Alessandro L et al. (2002). « A hybrid large vocabulary handwritten word recognition system using neural networks with hidden Markov models ». In : *ICFHR*, p. 99-104 (cf. p. 28).

Kölsch, Andreas et al. (2017). « Real-time document image classification using deep CNN and extreme learning machines ». In : *arXiv preprint arXiv :1711.05862* (cf. p. 27).

Kozielski, Michal et al. (2014). « Multilingual off-line handwriting recognition in real-world images ». In : *DAS*, p. 121-125.

Krizhevsky, Alex, Ilya Sutskever et Geoffrey E. Hinton (2012). « Imagenet classification with deep convolutional neural networks ». In : (cf. p. 27, 76).

Lavrenko, Victor, Toni M Rath et R Manmatha (2004). « Holistic word recognition for handwritten historical documents ». In : *DIAL*, p. 278-287 (cf. p. 80, 82).

Lemaitre, A, Jean Camillerapp et B Couasnon (2008). « Approche perceptive pour la reconnaissance de filets bruités, Application à la structuration de pages de journaux ». In : *Colloque International Francophone sur l'Ecrit et le Document*. Groupe de Recherche en Communication Ecrite, p. 61-66.

Lemaitre, Aurélie, Jean Camillerapp et Bertrand Coüasnon (2009). « Multi-script baseline detection using perceptive vision ». In : *14th Biennial Conference of the International Graphonomics Society (IGS 2009)* (cf. p. 24).

– (2011). « A perceptive method for handwritten text segmentation ». In : *Document Recognition and Retrieval XVIII*. T. 7874. International Society for Optics et Photonics, p. 78740C (cf. p. 24).

Lemaitre, Aurélie et Jean Camillerapp (2006). « Text line extraction in handwritten document with kalman filter applied on low resolution image ». In : *Document Image Analysis for Libraries, 2006. DIAL'06. Second International Conference on*. IEEE, 8-pp.

Likforman-Sulem, Laurence (2003). « Apport du traitement des images à la numérisation des documents manuscrits anciens ». In : *Document numérique* 7.3, p. 13-26.

Likforman-Sulem, Laurence, Abderrazak Zahour et Bruno Taconet (2007). « Text line segmentation of historical documents : a survey ». In : *International Journal of Document Analysis and Recognition (IJDAR)* 9.2-4, p. 123-138.

Likforman-Sulem, Laurence, Jérôme Darbon et Elisa H Barney Smith (2011). « Enhancement of historical printed document images by combining total variation regularization and non-local means filtering ». In : *Image and vision computing* 29.5, p. 351-363 (cf. p. 24).

Little, Greg et al. (2010). « Exploring Iterative and Parallel Human Computation Processes ». In : HCOMP '10, p. 68-76. doi : 10.1145/1837885.1837907. url : http://doi.acm.org/10.1145/1837885.1837907 (cf. p. 55).

Lladós, Josep et al. (2012). « On the influence of word representations for handwritten word spotting in historical documents ». In : *IJPRAI* 26.05, p. 1263002-1-25 (cf. p. 19, 44).

Lowe, David G (1999b). « Object recognition from local scale-invariant features ». In : *ICCV*, p. 1150-1157.

– (1999a). « Object recognition from local scale-invariant features ». In : *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. T. 2. Ieee, p. 1150-1157 (cf. p. 26).

– (2004). « Distinctive image features from scale-invariant keypoints ». In : *International journal of computer vision* 60.2, p. 91-110 (cf. p. 26).

Manmatha, Raghavan et Nitin Srimal (1999). « Scale space technique for word segmentation in handwritten documents ». In : *Scale-Space Theories in Computer Vision*. Springer, p. 22-33.

Manmatha, Raghavan, Chengfeng Han et Edward M Riseman (1996). « Word spotting : A new approach to indexing handwriting ». In : *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. IEEE, p. 631-637.

Marti, U-V et Horst Bunke (2001). « Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system ». In : *International journal of Pattern Recognition and Artificial intelligence* 15.01, p. 65-90 (cf. p. 25).

– (2002). « The IAM-database : an English sentence database for offline handwriting recognition ». In : *International Journal on Document Analysis and Recognition* 5.1, p. 39-46 (cf. p. 80).

Masci, Jonathan et al. (2011). « Stacked convolutional auto-encoders for hierarchical feature extraction ». In : *ICANN*, p. 52-59 (cf. p. 28).

Matsunaga, Andréa, Austin Mast et José A.B. Fortes (2016). « Workforce-efficient consensus in crowdsourced transcription of biocollections information ». In : *Future Generation Computer Systems* 56.C, p. 526-536. issn : 0167739X.

DOI : 10.1016/j.future.2015.07.004. URL : http://linkinghub.elsevier.com/retrieve/pii/S0167739X15002277 (cf. p. 52).

Mikolov, Tomas et al. (2013). « Distributed representations of words and phrases and their compositionality ». In : *Advances in neural information processing systems*, p. 3111-3119 (cf. p. 40).

Mioulet, Luc et al. (2015). « Language identification from handwritten documents ». In : *ICDAR*, p. 676-680 (cf. p. 36).

Mohamad, Muhammad'Arif et al. (2015). « A review on feature extraction and feature selection for handwritten character recognition ». In : *International Journal of Advanced Computer Science and Applications* 6.2, p. 204-212 (cf. p. 26).

Morillot, Olivier, Laurence Likforman-Sulem et Emmanuele Grosicki (2013). « New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks ». In : t. 22. 2, p. 023028-023028 (cf. p. 80).

Mouchere, Harold et al. (2017). « PiFF : a Pivot File Format ». In : (cf. p. 50).

Moysset, Bastien et al. (2014). « The A2iA multi-lingual text recognition system at the second Maurdor evaluation ». In : *ICFHR*, p. 297-302 (cf. p. 35, 77).

Murdock, Michael et al. (2015). « ICDAR 2015 competition on text line detection in historical documents ». In : *ICDAR*, p. 1171-1175.

Nair, Vinod et Geoffrey E Hinton (2010). « Rectified Linear Units Improve Restricted Boltzmann Machines ». In : *Proceedings of the 27th international conference on machine learning (ICML'10)*. Haifa, Israel, p. 807-814 (cf. p. 98).

Nakayama, Hideki et Noriki Nishida (2017). « Zero-resource machine translation by multimodal encoder–decoder network with multimedia pivot ». In : *Machine Translation* 31.1-2, p. 49-64 (cf. p. 19).

Nicolaou, Anguelos et Basilios Gatos (2009). « Handwritten text line segmentation by shredding text into its lines ». In : *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, p. 626-630.

Nicolas, Stephane, Thierry Paquet et Laurent Heutte (2004). « Un panorama des méthodes syntaxiques pour la segmentation d'images de documents manuscrits ». In : *8ème Colloque International Francophone sur l'Ecrit et le Document, CIFED'2004*, pp-237.

Nicolas, Stéphane et al. (2005). « Handwritten document segmentation using hidden Markov random fields ». In : *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE, p. 212-216.

Oprean, Cristina et al. (2013). « Using the Web to create dynamic dictionaries in handwritten out-of-vocabulary word recognition ». In : *ICDAR*, p. 989-993.

Oquab, Maxime et al. (2014). « Learning and transferring mid-level image representations using convolutional neural networks ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1717-1724 (cf. p. 43).

Otsu, Nobuyuki (1975). « A threshold selection method from gray-level histograms ». In : *Automatica* 11.285-296, p. 23-27 (cf. p. 24).

Ouwayed, Nazih et Abdel Belaïd (2012). « A general approach for multi-oriented text line extraction of handwritten documents ». In : *International Journal on Document Analysis and Recognition (IJDAR)* 15.4, p. 297-314.

Özsu, M. T. et Patrick Valudriez (1991). *Principles of Distributed Database Systems*. 562 p. Prentice-Hall.

Pan, Sinno Jialin et Qiang Yang (2010a). « A survey on transfer learning ». In : *IEEE Tran. on KDE* 22.10, p. 1345-1359 (cf. p. 19, 74).

– (2010b). « A survey on transfer learning ». In : *IEEE Transactions on knowledge and data engineering* 22.10, p. 1345-1359 (cf. p. 19).

Panichkriangkrai, Chulapong, Liang Li et Kozaburo Hachimura (2013). « Character segmentation and retrieval for learning support system of japanese historical books ». In : *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, p. 118-122.

Panzner, Maximilian et Philipp Cimiano (2016). « Comparing hidden markov models and long short term memory neural networks for learning action representations ». In : *International Workshop on Machine Learning, Optimization and Big Data*. Springer, p. 94-105 (cf. p. 81).

Papavassiliou, Vassilis et al. (2010). « Handwritten document image segmentation into text lines and words ». In : *Pattern Recognition* 43.1, p. 369-377.

Park, Hee-Seon et Seong-Whan Lee (1996). « Off-line recognition of large-set handwritten characters with multiple hidden Markov models ». In : *PR* 29.2, p. 231-244 (cf. p. 28).

Peters, Matthew E. et al. (2018). « Deep contextualized word representations ». In : (cf. p. 40).

Pham, Vu et al. (2014). « Dropout improves recurrent neural networks for handwriting recognition ». In : *ICFHR*, p. 285-290 (cf. p. 35, 85, 120).

Piotrowski, Michael (2012). « Natural language processing for historical texts ». In : *Synthesis Lectures on Human Language Technologies* 5.2, p. 1-157.

Pletschacher, Stefan et Apostolos Antonacopoulos (2010). « The PAGE (page analysis and ground-truth elements) format framework ». In : *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, p. 257-260 (cf. p. 50).

Pratikakis, Ioannis et al. (2014). « ICFHR 2014 competition on handwritten keyword spotting (H-KWS 2014) ». In : *ICFHR*, p. 814-819.

Pratikakis, Ioannis et al. (2016). « ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016) ». In : *Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR'16)*. Shenzhen, China, p. 619-623.

Pu, Yunchen et al. (2016). « Variational autoencoder for deep learning of images, labels and captions ». In : *Advances in neural information processing systems*, p. 2352-2360 (cf. p. 28).

Puigcerver, Joan, Alejandro H Toselli et Enrique Vidal (2015a). « ICDAR 2015 competition on keyword spotting for handwritten documents ». In : *ICDAR*, p. 1176-1180.

– (2015b). « ICDAR2015 Competition on Keyword Spotting for Handwritten Documents ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 1176-1180.

Qiu, Minghui et al. (2017). « Alime chat : A sequence to sequence and rerank based chatbot engine ». In : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*. T. 2, p. 498-503 (cf. p. 92).

Quraishi, Md Iqbal et al. (2013). « A novel hybrid approach to restore historical degraded documents ». In : *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on*. IEEE, p. 185-189 (cf. p. 24).

Rath, TM et al. (2002). « Indexing for a digital library of George Washington's manuscripts : a study of word matching techniques ». In : *CIIR Technical Report* (cf. p. 63).

Rath, Tony M et Rudrapatna Manmatha (2007). « Word spotting for historical documents ». In : *IJDAR* 9.2, p. 139-152 (cf. p. 82).

Razak, Zaidi et al. (2008). « Off-line handwriting text line segmentation : A review ». In : *International journal of computer science and network security* 8.7, p. 12-20.

Rehman, Amjad et Tanzila Saba (2011). « Document skew estimation and correction : analysis of techniques, common problems and possible solutions ». In : *Applied Artificial Intelligence* 25.9, p. 769-787 (cf. p. 25).

Rifai, Salah et al. (2011). « Contractive auto-encoders : Explicit invariance during feature extraction ». In : *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, p. 833-840 (cf. p. 28).

Romero, Verónica et al. (2013). « The ESPOSALLES database : An ancient marriage license corpus for off-line HWR ». In : *PR* 46.6, p. 1658-1669 (cf. p. 64).

Rothacker, Leonard et Gernot A Fink (2015). « Segmentation-free query-by-string word spotting with Bag-of-Features HMMs ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 661-665.

Rothfeder, Jamie L, Shaolei Feng et Toni M Rath (2003). « Using corner feature correspondences to rank word images by similarity ». In : *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*. T. 3. IEEE, p. 30-30.

Rusiñol, Marçal et al. (2015). « Efficient segmentation-free keyword spotting in historical document collections ». In : *Pattern Recognition* 48.2, p. 545 -555. issn : 0031-3203. doi : http://dx.doi.org/10.1016/j.patcog.2014.08.021. url : http://www.sciencedirect.com/science/article/pii/S0031320314003355.

Saha, Satadal et al. (2010). « A Hough transform based technique for text segmentation ». In : *arXiv preprint arXiv :1002.4048*.

Sanchez, Joan Andreu et al. (2017). « ICDAR2017 Competition on Handwritten Text Recognition on the READ Dataset ». In : *Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR'17)*. Kyoto, Japan, p. 1383-1388.

Schuster, Mike et Kuldip K Paliwal (1997). « Bidirectional recurrent neural networks ». In : *IEEE Transactions on Signal Processing* 45.11, p. 2673-2681 (cf. p. 33).

Senior, Andrew W et Anthony J Robinson (1998). « An off-line cursive handwriting recognition system ». In : *PAMI* 20.3, p. 309-321.

Senior, Andrew William (1994). « Off-line cursive handwriting recognition using recurrent neural networks ». Thèse de doct. England.

Sermanet, Pierre et al. (2013). « OverFeat : Integrated Recognition, Localization and Detection using Convolutional Networks ». In : *CoRR* abs/1312.6229. url : http://arxiv.org/abs/1312.6229.

Seuret, Mathias et al. (2015). « Clustering historical documents based on the reconstruction error of autoencoders ». In : *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*. ACM, p. 85-91 (cf. p. 28).

Shi, Zhixin et Venu Govindaraju (2004). « Historical document image enhancement using background light intensity normalization ». In : *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. T. 1. IEEE, p. 473-476 (cf. p. 24).

Simistira, Fotini et al. (2015). « Recognition of historical Greek polytonic scripts using LSTM networks ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 766-770.

Simonyan, Karen et Andrew Zisserman (2014a). « Very deep convolutional networks for large-scale image recognition ». In : *arXiv preprint arXiv :1409.1556* (cf. p. 27, 76).

– (2014b). « Very Deep Convolutional Networks for Large-Scale Image Recognition ». In : *CoRR* abs/1409.1556. url : http://arxiv.org/abs/1409.1556.

Stamatopoulos, Nikolaos et al. (2013). « Icdar 2013 handwriting segmentation contest ». In : *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, p. 1402-1406.

Su, Bolan, Shijian Lu et Chew Lim Tan (2010). « Binarization of historical document images using the local maximum and minimum ». In : *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, p. 159-166 (cf. p. 24).

Sun, Chen et al. (2015). « Temporal localization of fine-grained actions in videos by domain transfer from web images ». In : *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, p. 371-380 (cf. p. 43).

Suryani, Dewi, Patrick Doetsch et Hermann Ney (2016). « On the Benefits of Convolutional Neural Network Combinations in Offline Handwriting Recognition ». In : *ICFHR*, p. 193-198.

Sutskever, Ilya, Oriol Vinyals et Quoc V Le (2014). « Sequence to sequence learning with neural networks ». In : *Advances in neural information processing systems*, p. 3104-3112 (cf. p. 112).

Swaileh, Wassim et al. (2017a). « Handwriting Recognition with Multigrams ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 137-142 (cf. p. 25).

– (2017b). « Handwriting Recognition with Multigrams ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 137-142 (cf. p. 39).

Szegedy, Christian et al. (2014). « Going Deeper with Convolutions ». In : *CoRR* abs/1409.4842. url : http://arxiv.org/abs/1409.4842.

Tensmeyer, Chris et Tony Martinez (2017). « Document image binarization with fully convolutional neural networks ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 99-104 (cf. p. 27).

Tensmeyer, Chris, Daniel Saunders et Tony Martinez (2017). « Convolutional Neural Networks for Font Classification ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 985-990 (cf. p. 27).

Terasawa, Kengo et Yuzuru Tanaka (2009). « Slit style HOG feature for document image word spotting ». In : *ICDAR*, p. 116-120 (cf. p. 26, 75).

Thomas, S. et al. (2015). « A deep HMM model for multiple keywords spotting in handwritten documents ». In : *Pattern Analysis and Applications* 18.4, p. 1003-1015.

Toledo, Juan Ignacio et al. (2016). « Election Tally Sheets Processing System ». In : *DAS*, p. 364-368 (cf. p. 75).

Uchida, Seiichi, Eiji Taira et Hiroaki Sakoe (2001). « Nonuniform slant correction using dynamic programming ». In : *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, p. 434-438 (cf. p. 25).

Vania, Clara et Adam Lopez (2017). « From Characters to Words to in Between : Do We Capture Morphology ? » In : *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. Vancouver, Canada, p. 2016-2027. url : http://aclweb.org/anthology/P17-1184 (cf. p. 95, 116).

Vincent, Pascal et al. (2008). « Extracting and composing robust features with denoising autoencoders ». In : *Proceedings of the 25th international conference on Machine learning*. ACM, p. 1096-1103.

Vincent, Pascal et al. (2010). « Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion ». In : *Journal of machine learning research* 11.Dec, p. 3371-3408 (cf. p. 28).

Vinciarelli, Alessandro et Samy Bengio (2002). « Offline cursive word recognition using continuous density hidden markov models trained with PCA or ICA features ». In : *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. T. 3. IEEE, p. 81-84 (cf. p. 26).

Vinciarelli, Alessandro et Juergen Luettin (1999). *Off-line cursive script recognition based on continuous density HMM*. Rapp. tech. IDIAP (cf. p. 25).

– (2001). « A new normalization technique for cursive handwritten words ». In : *Pattern recognition letters* 22.9, p. 1043-1050 (cf. p. 24).

Vinyals, Oriol et al. (2015). « Show and tell : A neural image caption generator ». In : *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, p. 3156-3164 (cf. p. 92).

Voigtlaender, Paul, Patrick Doetsch et Hermann Ney (2016a). « Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks ». In : *ICFHR*, p. 2228-233.

– (2016b). « Handwriting recognition with large multidimensional long short-term memory recurrent neural networks ». In : *Proc. of ICFHR*, p. 228-233.

Wang, Cong, Fei Yin et Cheng-Lin Liu (2017). « Scene Text Detection with Novel Superpixel Based Character Candidate Extraction ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 929-934 (cf. p. 27).

Werbos, Paul J (1990). « Backpropagation through time : what it does and how to do it ». In : *Proceedings of the IEEE* 78.10, p. 1550-1560 (cf. p. 32).

Wigington, Curtis et al. (2017). « Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network ». In : *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, p. 639-645 (cf. p. 27).

Wu, Yi-Chao, Fei Yin et Cheng-Lin Liu (2015). « Evaluation of neural network language models in handwritten Chinese text recognition ». In : *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, p. 166-170 (cf. p. 40).

– (2017). « Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models ». In : *Pattern Recognition* 65, p. 251-264 (cf. p. 40).

Xie, Zecheng et al. (2016). « Fully Convolutional Recurrent Network for Handwritten Chinese Text Recognition ». In : *arXiv preprint arXiv :1604.04953* (cf. p. 36).

Yi, Xiaohan et al. (2017). « CNN Based Page Object Detection in Document Images ». In : *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. T. 1. IEEE, p. 230-235 (cf. p. 27).

Yousfi, Sonia, Sid-Ahmed Berrani et Christophe Garcia (2017). « Contribution of recurrent connectionist language models in improving LSTM-based Arabic text recognition in videos ». In : *Pattern Recognition* 64, p. 245-254.

Zamora-Martinez, Francisco et al. (2014). « Neural network language models for off-line handwriting recognition ». In : *Pattern Recognition* 47.4, p. 1642-1652 (cf. p. 40).

Zeeuw, Frank de, Axel Brink et Tijn van der Zant (2006). « Slant Correction using Histograms ». In : (cf. p. 25).

Zhang, Zheng et al. (2016). « Multi-Oriented Text Detection with Fully Convolutional Networks ». In : *arXiv preprint arXiv :1604.04018*.

Zheng, Liang et al. (2016). « Good practice in CNN feature transfer ». In : *arXiv preprint arXiv :1604.00133* (cf. p. 28).

Zhou, Bolei et al. (2014). « Learning Deep Features for Scene Recognition using Places Database ». In : *Advances in Neural Information Processing Systems 27*. Sous la dir. de Z. Ghahramani et al. Curran Associates, Inc., p. 487-495. url : http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

Titre : Extraction d'information dans des documents manuscrits anciens

Mots clés : Reconnaissance d'écriture manuscrite, Apprentissage par transfert de connaissances, Réseaux de neurones, Documents historiques, Modèle optique, Modèle linguistique

Résumé : La tâche d'exploration dans des ressources inexploitées mais nouvellement numérisées, afin d'y trouver des informations pertinentes, est complexifiée par la quantité de ressources disponibles. Grâce au projet ANR CIRESFI, la ressource la plus importante, pour la Comédie-Italienne du XVIIIe siècle, est un ensemble de registres comptables constituée de 28 000 pages. L'extraction d'informations est un processus long et complexe qui demande une expertise à chaque étape : détection et segmentation, extraction de caractéristiques, reconnaissance d'écriture manuscrite. Les systèmes à base de réseaux de neurones profonds dominent dans l'ensemble ces approches. Le problème majeur est qu'ils nécessitent d'avoir une grande quantité de données pour réaliser leur apprentissage. Cependant, les registres de la Comédie-Italienne ne possèdent pas de vérité terrain. Pour palier ce manque de données, nous explorons des approches pouvant opérer un apprentissage par transfert de connaissance. Cela signifie utiliser un ensemble de données déjà étiquetées et disponibles, possédant un minimum de points communs avec nos données pour entraîner les systèmes, pour ensuite les appliquer sur nos données. L'ensemble de nos expérimentations nous ont montré la difficulté de réaliser cette tâche, chaque choix à chaque étape ayant un impact fort sur la suite du système. Nous convergeons vers une solution séparant le modèle optique du modèle de langage afin de réaliser un apprentissage indépendant avec différents types de ressources disponibles et se rejoignant grâce à une projection de l'ensemble des informations dans un espace commun non-latent.

Title : Extracting information in old handwritten documents

Keywords : Handwriting recognition, Transfer learning, Neural network, Historical documents, Optical model, Linguistic model

Abstract : Exploring unexploited but newly digitized resources to find relevant information is a complicated task due to the amount of available resources. Thanks to the ANR project CIRESFI, the most important resource for the Italian Comedy of the 18th century, is a set of accounting registers consisting of 28,000 pages. Information retrieval is a long and complex process that requires expertise at every step: detection and segmentation in paragraphs, lines or words, features extraction, handwriting recognition. Systems based on deep neural networks dominate these approaches. The major issue is the need of a large amount of data to achieve their learning. However, the registers of the Italian Comedy have no ground truth. To overcome this lack of data, we explore approaches that involving transfer learning. That means using heterogeneous labeled and available data, with at least one common feature with our data to drive the systems, and then applying them to our data. All of our experiments have shown us the difficulty of carrying out this task, each choice at each stage having a strong impact on the rest of the system. We converge on a solution separating the optical model from the language model in order to achieve independent learning with different available resources and joining together thanks to a projection of the information into a non-latent common space.