Cascade de réseaux BLSTM vérifiée par le lexique pour la reconnaissance d'écriture

Bruno STUNER[1,2] Clément CHATELAIN[1,3] Thierry PAQUET[1,2]

[1]Laboratoire LITIS EA 4108, Normandie Université, [2]Université de Rouen, [3]INSA de Rouen, France

prenom.nom@litislab.eu

Résumé

La reconnaissance de l'écriture manuscrite est une tâche difficile alliant le traitement d'image et le traitement de la langue. Récemment des modèles de réseaux de neurones récurrents à base de LSTM ont permis des progrès significatifs dans ce domaine. Ces réseaux sont généralement couplés avec des connaissances lexicales et linguistiques au moment du processus de décodage afin de corriger les erreurs de reconnaissance au niveau caractère, typiquement à l'aide d'un décodage dirigé par le lexique. Pourtant les performances élevées des réseaux LSTM laissent entrevoir la possibilité de les utiliser en s'affranchissant de ce type de processus. Nous proposons dans cet article un décodage sans lexique que nous couplons à une méthode de vérification par le lexique. Cette méthode de contrôle par le lexique présente des propriétés intéressantes et nous permet de combiner efficacement des réseaux LSTM en les mettant en cascade. Cette approche permet d'accélérer le décodage, tout en étant peu sensible au changement de taille de lexique. Notre approche présente des résultats prometteurs, permettant d'avoir une erreur faible, en concédant une part de rejets. Ces rejets peuvent être finalement traités par un décodage classique dirigé par le lexique, permettant de nous placer proches des meilleures méthodes à l'état de l'art.

Mots Clef

Cascade de classifieurs, Réseaux de neurones récurrents, LSTM, reconnaissance d'écriture manuscrite.

Abstract

Handwritten word recognition is a tough task, mixing image and natural language processing. Recently new recurrent neural networks with LSTM cells allowed significant improvements in this field. These networks are generally copled with lexical and linguistic knowledges in order to correct character misrecognitions, namely using a lexicon driven decoding. Yet the high performances of LSTM networks let us think that there is a room to use them without lexical decoding. In this article we propose a lexicon-free decoding, combined with a lexicon verification method. This lexicon control method presents some interesting properties and enables us to efficiently combine LSTM networks in a cascade framework. This cascade process is not driven by the lexicon but simply controled, allowing it to speed up the decoding while being nearly insensitive to the lexicon size. Our approach presents promising results with à low error rate by conceding rejects. Those rejects can finally be processed by a standard lexical decoding, enabling us to get close to state of the art methods.

Keywords

Cascade of classifiers, Recurrent neural networks, LSTM, Handwritten word recognition.

1 Introduction

Actuellement, les systèmes de reconnaissance de l'écriture manuscrite ou de la parole procèdent par exploration des solutions en étant guidés par des ressources linguistiques telles qu'un lexique et/ou un modèle de langage. Cependant, la reconnaissance dirigée par le lexique pose de nombreux problèmes, notamment en temps et précision, par exemple sur un lexique de grande taille tel que le dictionnaire Français-Gutenberg qui comporte 336 531 mots. L'utilisation d'un lexique est par ailleurs néfaste lorsque l'on soumet un élément à un lexique qui ne le contient pas puisque le décodage propose le mot qui s'en approche mais qui n'est pas la bonne solution, même si les caractères ont été correctement reconnus. Un lexique non adapté dégrade alors les performances du système.

Récemment, des modèles de réseaux de neurones récurrents à base de cellules de mémoire à long et court terme (LSTM)[10] ont permis des progrès significatifs de la reconnaissance de l'écriture cursive, notamment grâce a une nouvelle méthode d'apprentissage, la classification temporelle connexionniste (CTC) [9]. Ces réseaux sont généralement couplés à un étage de décodage dirigé par le lexique, permettant de trouver des solutions cohérentes en corrigeant les éventuelles erreurs de reconnaissance de caractères. Ces techniques ont permis d'obtenir d'excellents résultats lors des compétitions [11], [4].

Dans cet article et contrairement aux approches précédentes, nous proposons d'explorer une nouvelle méthode de reconnaissance n'introduisant pas d'étape de décodage

dirigée par le lexique. Nous tentons de combiner les performances des réseaux LSTM et leur propriétés observées au cours de leur apprentissage pour construire un système de décision avec rejet procédant par vérification lexicale. Le rejet étant un mécanisme capital dans les systèmes où les erreurs coûtent cher, comme pour la lecture d'adresse, cette nouvelle approche permet une réduction significative de l'erreur par le rejet. Cette méthode permet également de se dégager des problèmes liés à la taille du lexique, réduisant significativement sa sensibilité à ce paramètre, tout en étant plus rapide qu'une reconnaissance dirigée par le lexique.

L'intérêt d'un système de décision par vérification lexicale réside dans sa capacité à générer facilement du rejet, car il est très rare qu'un système génère une hypothèse présente dans le lexique par erreur. Cependant, pour qu'un système de reconnaissance sans lexique atteigne les performances de l'état de l'art il faut qu'il soit capable de reconnaître sans erreur tous les caractères manuscrits présents dans le mot. C'est ce second point qui est très difficile à réaliser. Pour y parvenir, nous proposons dans cet article une stratégie de mise en cascade de BLSTM complémentaires combinant un système de rejet par vérification lexicale.

Cet article est organisé de la manière suivante : Dans un premier temps, nous rappelons les principes des méthodes de reconnaissance dirigées par le lexique, ainsi que des réseaux récurrents. Ensuite, nous présentons le principe de cascade de classifieurs et la façon dont nous l'exploitons pour réaliser une cascade de réseaux récurrents LSTM intégrant un système de décision par vérification lexicale. Finalement, les résultats de nos expérimentations sur la base de mots isolés Rimes sont présentés dans la dernière section.

2 Travaux antérieurs

2.1 Reconnaissance de l'écriture et décodage par le lexique

Il existe de nombreuses méthodes pour la reconnaissance de l'écriture [18] selon qu'on considère une segmentation explicite ou implicite, selon les caractéristiques extraites, ou encore selon le type de classifieur mis en œuvre pour la reconnaissance de caractère (classifieurs discriminants dans le cas des approches hybrides [22] ou gaussiennes dans le cas des modèles de Markov cachés (HMM) [5]). Le point commun à l'ensemble des approches est qu'elle réalisent toute un décodage *dirigé par le lexique*. En effet, comme les performances des systèmes de reconnaissance des caractères manuscrits cursifs sont insuffisantes, la décision de reconnaissance au niveau mot est généralement différée à la fin du processus en explorant la meilleure combinaison d'hypothèses de caractères formant un mot d'un lexique.

Les méthodes à l'état de l'art pour la reconnaissance dirigée par le lexique se basent sur les modèles de Markov cachés (HMM [20]) ; un panorama leur a d'ailleurs été dédié [19]. La force des HMM repose sur la modélisation sans segmentation explicite des caractères, ce qui permet de laisser le processus de reconnaissance réaliser la segmentation. Le second aspect intéressant avec les HMM est qu'ils intègrent aisément des modèles statistiques de langages (n-gram de mots), à condition toutefois de définir un lexique de travail. Cependant la taille du lexique est problématique, comme il est montré dans [15], étude dans laquelle un vocabulaire est considéré de grande taille à partir de 1000 mots.

L'un des premiers problèmes posé par la taille du lexique est son influence sur la précision du modèle. Dans [11] qui rapporte les résultats de la compétition RIMES 2009, seul le système proposé par TUM ne se base pas sur des HMM. Pour les autres systèmes reposant sur des modèles HMM on peut voir très nettement l'influence de la taille du lexique (respectivement de 100, 1612 et 5334 mots) sur la performance de reconnaissance. Par exemple, le système du LITIS enregistre une différence de performances de 7.45% entre les deux plus grands vocabulaires, contre à peine 2.15% pour TUM. La taille du lexique a aussi un rôle prépondérant dans l'utilisation de ces méthodes pour des problèmes réels où le lexique n'est plus lié à la nature d'une base mais plutôt à une application et à une langue. Dans certains cas l'intégration de ressources linguistiques pour améliorer les performances n'est plus compatible avec des contraintes de temps réel.

Il existe néanmoins de nombreuses méthodes permettant d'élaguer le nombre de mots candidats d'un lexique au cours du processus de reconnaissance [15]. Ces méthodes présentent toutefois d'autres problèmes comme la perte de solutions lorsque l'élagage est trop sévère, conduisant à une grande remontée d'erreurs.

Certaines méthodes utilisant les modèles de Markov cachés permettent un décodage sans lexique [2], mais ces dernières sont complexes et ne sont pas encore au niveau de l'état de l'art, ni utilisables pour des problèmes concrets du fait du manque de données pour les modéliser. Dernièrement, des méthodes se basant sur la composition du lexique à partir de suffixe et de préfixe de la langue, modélisés par des n-grams présentent des résultats à l'état de l'art pour la reconnaissance de mot hors lexique [13].

Les modèles HMM sont désormais couplés à des réseaux LSTM, car ils présentent des performances pures très élevées en reconnaissance de caractères cursifs. Cependant nous nous demandons si de tels réseaux ne pourraient pas se passer de modèles dirigés par le lexique. Nous rappelons brièvement leur principe dans le paragraphe qui suit.

2.2 Réseau de neurones récurrents LSTM

Les réseaux récurrents standards proposés il y a près de 30 ans tirent leur force de leur capacité de mémoire pour le traitement de séquence, grâce à des connexions récurrentes apportant le contexte de l'élément précédant dans la séquence. Cependant les réseaux récurrents souffrent du problème de disparition du gradient, les empêchant d'ap-

prendre sur des durées longues. Les blocs de mémoire à long et court terme (LSTM) ont alors été introduits par Hochreiter [14] afin de dépasser cette limitation.
La cellule LSTM permet d'apprendre des données à long ou à court terme lors du traitement de séquences grâce à un mécanisme de portes (entrée, oubli et sortie) activées par une sigmoïde, et commandées par une cellule multiplicatrice. La mise en mémoire se faisant uniquement dans le sens de la séquence, les RNN bidirectionnels ont été introduits [21]. Cette idée a ensuite été étendue aux réseaux LSTM [9] pour donner naissance aux réseaux récurrents bidirectionnels à cellules LSTM : les réseaux BLSTM.
Cependant les réseaux LSTM n'ont connu leur essor que suite à l'introduction d'une nouvelle méthode d'apprentissage, la classification temporelle connexionniste (CTC)[6], qui est en fait une variante de l'algorithme Forward Backward mis en œuvre pour l'apprentissage des HMM. La classification temporelle connexionniste (CTC) participe grandement à la possibilité d'apprendre de tels réseaux, notamment avec l'introduction d'une classe "joker" ou "blanc".
Avec un apprentissage CTC et le joker, les réponses en sortie d'un réseau BLSTM forment des pics très marqués, c'est-à-dire que la probabilité d'une classe est très forte (supérieure à 0.95) et les autres très faibles. Lorsque le réseau ne détecte pas de caractère, la classe joker a donc une probabilité très élevée. Un exemple de sorties d'un BLSTM pour le mot "vouloir" est donné figure 1, pour des raisons de clartés dans la légende, seul les lettres minuscules sont affichées et les lettres du mots sont mises en couleurs.

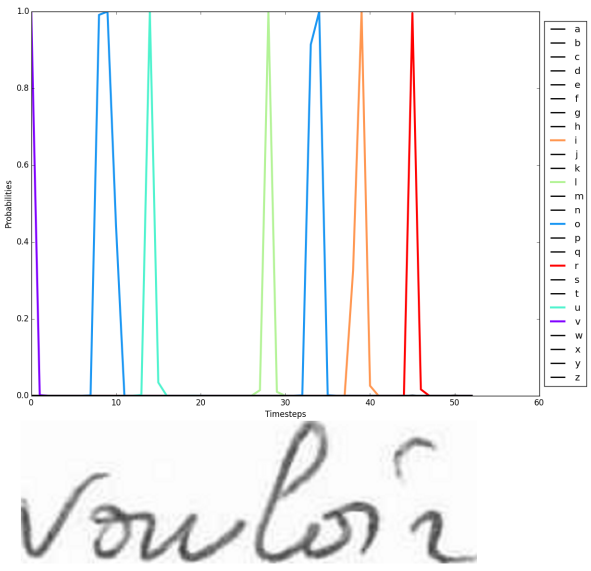

FIGURE 1 – Sorties d'un BLSTM à l'issue d'un apprentissage CTC

Ces sorties en pic permettent notamment l'application d'un décodage simple sans lexique ("Best path decoding" dans [8]) en prenant la classe de probabilité maximale de chaque élément de la séquence ou trame, et en supprimant toutes les répétitions successives de chaque classe (joker compris), puis les "joker".
Les performances à l'issue de ce décodage sans lexique sur une tâche de reconnaissance mot sont très intéressantes et n'ont à notre connaissance pas motivé d'études plus approfondies pour tenter de les exploiter au mieux. C'est dans cette perspective que nous proposons d'accroître les performances des architectures à base de réseaux BLSTM dans une tâche de reconnaissance d'écriture vérifiée par le lexique. Une piste intéressante est l'utilisation d'un structure en cascade qui a donné de bons résultats dans d'autres domaines de la reconnaissance de formes.

3 Cascade de réseaux LSTM

3.1 Cascade de classifieurs

La cascade de classifieurs est une méthode de combinaison se basant sur le passage d'objets à reconnaître à travers plusieurs classifieurs complémentaires, afin d'affiner des résultats de reconnaissance. Le cœur du fonctionnement de la cascade est un système décisionnel permettant le rejet. Souvent le critère de rejet consiste à appliquer un seuil sur la confiance du classifieur à l'étage courant de la cascade. Ce mécanisme de rejet dans la cascade est essentiel et permet d'accélérer les temps de traitement. Pour être efficace, la complémentarité des classifieurs de la cascade reste cependant primordiale. Plusieurs façons de mettre en œuvre le principe de cascade ont été proposées dans la littérature, permettant de traiter différents problèmes et notamment des problématiques de reconnaissance de formes.
L'article le plus connu concernant l'utilisation de cascade de classifieurs est celui de Viola et Jones [23], pour la détection de visage. Ici la cascade se base sur un grand ensemble de classifieurs faibles et différents mais permettant un traitement rapide avec un fort rejet. L'image est analysée morceau par morceau, l'imagette doit passer l'ensemble des classifieurs sans se faire rejeter pour être reconnue comme contenant l'objet d'intérêt.
Dans [25] et [26], le principe de cascade est utilisé pour combiner les résultats d'ensemble de classifieurs forts et différents dans leurs architectures ou leurs caractéristiques d'entrée. Ces ensembles reconnaissent un nombre important d'objets avec une erreur faible, tout en s'appuyant sur un système de décision leur permettant de transmettre des éléments rejetés aux classifieurs suivants.
C'est ce dernier principe qui a retenu notre attention, en effet les réseaux LSTM semblent posséder toutes les caractéristiques d'un classifeur fort. Nous devons alors trouver un système décisionnel cohérent avec la mise en cascade et les propriétés des réseaux LSTM.

3.2 Système de décision par vérification lexicale

L'un des aspects important dans la mise en œuvre d'une cascade de classifieurs réside dans l'étage de décision chargé d'accepter ou rejeter une hypothèse. Dans notre cas

où nous cherchons à reconnaître des séquences de caractères quelconques, nous avons adopté une stratégie de vérification par un lexique.
Cette vérification par un lexique consiste à accepter la chaîne de caractères proposée à la sortie d'un réseau BLSTM si elle appartient au dictionnaire, et à la rejeter sinon. La chaîne de caractères est le résultat du décodage "Best path decoding" [8] détaillé précédemment (2.2).
Remarquons qu'il est peu probable qu'une séquence mal reconnue soit présente dans le lexique. Nous pouvons formuler cette probabilité à l'aide de l'équation suivante :

$$1 - (1 - CER)^n \times \frac{m_n}{d^n} \quad (1)$$

$1 - (1 - CER)^n$ représente la probabilité de réaliser une erreur de reconnaissance sur un mot en fonction de son nombre de caractère, où CER est le taux d'erreur caractère et n le nombre de caractère. $\frac{m_n}{d^n}$ est la probabilité qu'un mot appartiennent à un dictionnaire, où m_n est le nombre de mots de taille n dans le dictionnaire et d^n le nombre de séquences possibles de caractères de longueur n lorsqu'il y a d classes de caractères.
En calculant cette probabilité pour un mot de 3 caractères, pour une erreur de reconnaissance caractère de 0.15 (supérieure à celle du BLSTM), et en prenant le dictionnaire des mots de la compétition RIMES 2011 où $m_3 = 233$, nous obtenons la probabilité de 0.016% de se tromper et d'être dans le dictionnaire. Cette probabilité décroit de manière exponentielle avec la longueur du mot, on peut donc bien affirmer qu'*il est peu probable qu'une séquence mal reconnue soit présente dans le lexique*.
En réalisant une première expérience (première ligne du tableau 1) afin de tester notre système de décision, nous constatons que celui-ci permet bien de rejeter mais il réalise tout de même quelques erreurs en acceptant à tort une faible proportion d'hypothèses qui sont présentes dans le lexique. Parmi ces erreurs, il y a principalement des erreurs de casse, d'accentuations et de pluriels. Disposant d'une méthode simple et performante de rejet, nous montrons maintenant comment des réseaux BLSTM peuvent être efficacement combinés en cascade, en s'appuyant sur la complémentarité de plusieurs réseaux.

3.3 Schéma de la cascade

L'idée générale de la cascade est de faire traiter les rejets d'un classifieur par l'étage suivant. Pour que cette méthode fonctionne efficacement, nous devons donc disposer de systèmes de classification complémentaires.
Une première idée de complémentarité aurait été d'apprendre les réseaux à partir des rejets du réseau précédent, hors plus il y aura de réseaux en cascade et moins il y aura d'exemples à apprendre, rendant difficile l'apprentissage et la convergence des réseaux. De plus ces réseaux deviendront spécialisés et n'apporteront rapidement plus de diversité, nous nous orientons donc vers une solution plus simple et tout aussi efficace.
Des expériences ont montré que des réseaux récurrents à cellules LSTM similaires ayant pour unique différence l'initialisation aléatoire des poids peuvent être combinés avec succès [16]. Ces réseaux ont des taux de lecture similaires mais les poids des connexions sont différents. C'est en nous inspirant de ces constatations qui illustrent une propriété spécifique des architectures BLSTM, que nous avons fait le choix d'exploiter la complémentarité de réseaux BLSTM initialisés de manières différentes.
Dans les expérimentations qui suivent nous proposons une cascade à trois niveaux basée sur la complémentarité de trois réseaux identiques, avec des initialisations aléatoires des poids différentes. La figure 2 reprend tous les éléments associés à la cascade, les trois réseaux et le module de décision. Elle présente également l'évolution d'une base d'images testée en fonction du niveau dans la cascade.

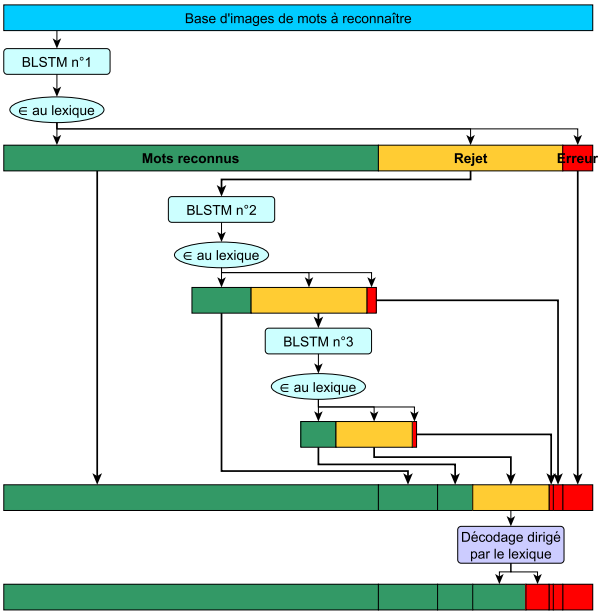

FIGURE 2 – Cascade de BLSTM

4 Expérimentations et résultats

4.1 Architecture de réseau sélectionnée

Comme annoncé précédemment, notre schéma de cascade présente 3 réseaux BLSTM basés sur la même architecture. L'architecture des réseaux BLSTM retenue est identique à celle utilisée dans [17]. Il s'agit d'un réseau à deux couches LSTM contenant respectivement 70 et 120 neurones, séparés par une couche de sous échantillonnage de 100 neurones cachés sans biais avec pour fonction d'activation une tangente hyperbolique. Le réseau est également composé de deux couches de réduction du nombre de trames. La première en entrée concatène les vecteurs des trames deux à deux, et la seconde à la sortie de la première couche concatène les trames précédentes deux à deux. Par exemple, si en entrée il y a 12 trames de largeur 1 pixel, il y aura 3 sorties

correspondant à 3 trames de largeur 4 pixels.
En entrée de notre réseau nous utilisons les histogrammes de gradient orientés (HOG) [3] qui se sont montrés efficaces pour la reconnaissance de l'écriture [1]. Pour utiliser ces caractéristiques, la hauteur de l'image est normalisée à $2^6 = 64$ pixels. Une fenêtre glissante de largeur 8 extrait les caractéristiques HOG avec un pas de 1 pixel.
Cette architecture a été retenue par son équilibre car elle présente à la fois des performances légèrement supérieures à l'architecture de référence [10] et elle permet aussi un décodage rapide des images, en moyenne 30 millisecondes. Des expériences nous ont également montré qu'avec le même ordre de paramètres pour un réseau, les performances variaient peu.
Les apprentissages sont réalisés avec la RNNLIB [7]. Nous réalisons 3 apprentissages obtenant des réseaux aux performances (erreur de reconnaissance des caractères) très similaires : 11.85%, 11.81% et 11.88%. Ces résultats étant très proches, l'ordre des réseaux n'a pas d'importance.

4.2 Décodage dirigé par le lexique

Dans le but de comparer nos résultats aux méthodes à l'état de l'art et afin de s'assurer de l'évolution positive de nos résultats lors de la cascade, nous utilisons en sortie de cascade un décodage simple dirigé par le lexique.
Pour réaliser le décodage dirigé par le lexique, nous avons retiré les probabilités de la classe joker ainsi que toutes les trames ayant pour probabilité maximale la classe joker, du fait que cette classe n'est pas représentée dans le lexique. Une fois les trames élaguées, nous réalisons un décodage de type Viterbi[24].

4.3 Base de données Rimes

Nous utilisons la base publique de mots isolés Rimes utilisée à l'occasion de la compétition ICDAR 2011 sur la reconnaissance de l'écriture française [12]. Cette base est divisée en trois parties, apprentissage (51737 images), validation (7464 images) et test (7776 images). On peut distinguer deux tâches : la tâche WR2 où le lexique est composée des 1692 mots de la base de test, et la tâche WR3 où le lexique contient les 5744 mots de la base.

4.4 Expérience préliminaire sans cascade

Cette expérience préliminaire a pour objectif de présenter les effets de notre système de décision appliqué à la sortie d'un BLSTM. L'architecture de BLSTM est celle sélectionnée pour la cascade et nous utilisons le décodage dirigé par le lexique permettant à la fois de nous comparer aux résultats à l'état de l'art mais également de nous assurer que notre système décisionnel ne détériore pas les résultats. Cette expérience met en jeu un réseau BLSTM, en sortie duquel trois stratégies sont comparées :

— La vérification lexicale ;
— Le décodage dirigé par le lexique ;
— La vérification lexicale suivie du décodage dirigé par le lexique.

Réseau	**Lecture**	**Erreur**	**Rejet**
BLSTM + vérification	66.37	**2.25**	31.38
BLSTM + lexique	88.35	11.65	0
BLSTM + vérification + lexique	88.72	11.28	0

TABLE 1 – Résultats préliminaires sur un réseau

Les résultats sont présentés dans le tableau 1. L'élément le plus remarquable est la très faible erreur obtenue grâce à la règle de vérification dans le lexique, elle n'est que de 2.25%, réduisant ainsi l'erreur de 80% par rapport à une méthode dirigée par le lexique. Cependant ce système nous laisse 31.38% de rejets qu'il faut réussir à traiter.
Il apparaît que le décodage effectué en sortie du mécanisme de vérification n'impacte que peu (différence de 0.37 points) les performances avec décodage dirigé par le lexique. L'intérêt est de n'effectuer un décodage par lexique que pour les 31% des mots que l'on n'a pas trouvé dans le lexique, divisant le temps de décodage dirigé par le lexique par 3. Cette expérience met bien en évidence les résultats prévus pour tenter de tirer le bénéfice d'une mise en cascade de classifieur. L'erreur est faible car peu de séquences faussement reconnues sont présentes dans le lexique et nous avons une quantité importante de rejets, tout en ayant de bons scores de lecture du fait de l'utilisation d'un BLSTM. Le principe de cascade peut alors tout à fait s'appliquer à cette situation, c'est ce que nous examinons dans la section suivante.

4.5 Résultats de la cascade

À partir des éléments précédents, nous testons donc le schéma de cascade complet sur la base Rimes. Nous présentons 4 tableaux de résultats, avec et sans décodage dirigé par le lexique et pour les tâches WR2 et WR3. Tous les taux présentés sont calculés sur l'ensemble de la base, à l'issue de chaque étage. La somme du taux de lecture, d'erreur et de rejet vaut 100, la mauvaise classification, MC, représente l'erreur faite sur les mots classifiés de l'étage uniquement.

Réseau	**Lecture**	**Erreur**	**Rejet**	**MC**
1er étage	66.37	2.25	31.38	3.3
2ème étage	74.05	2.59	23.37	4.2
3ème étage	78.07	2.82	19.11	5.4

TABLE 2 – Résultats de la cascade pour la tâche WR2

Réseau	**Lecture**	**Erreur**	**Rejet**
1er étage	91.18	8.82	0
2ème étage	92.23	7.77	0
3ème étage	92.85	7.15	0

TABLE 3 – Résultats de la cascade pour la tâche WR2 avec décodage dirigé par le lexique en sortie de chaque étage

Les tableaux 2 et 3 présentent les résultats obtenus pour la tâche WR2, c'est-à-dire pour un lexique de 1692 mots. On peut d'abord noter la bonne complémentarité de nos réseaux. En effet à l'issue du deuxième réseau BLSTM, le système a classés 25% des rejets en n'augmentant l'erreur que de 0.34%, soit seulement 4.2% d'erreur dans l'ensemble des rejets nouvellement classés.
Il faut ensuite noter que l'erreur de classification de 4.2% est plus importante que 3.3% qui est l'erreur du premier réseau, cependant cette erreur semble faible du fait qu'elle concerne un étage plus élevé dans la cascade. En effet plus les mots à reconnaître continuent de parcourir les classifieurs de la cascade plus ces mots peuvent être considérés comme difficiles à reconnaître.
Cette complémentarité se confirme à la sortie du troisième BLSTM, avec pas moins de 18% des rejets précédents classés, pour une erreur supplémentaire de 0.23%, avec toujours une mauvaise classification en hausse à 5.4%, qui reste cohérente avec la remarque précédente.
Nous pouvons conclure que la cascade semble bénéficier du système décisionnel et de la complémentarité des réseaux. Cette première conclusion est appuyée par les bons résultats d'un décodage dirigé par le lexique appliqué sur les rejets en sortie de cascade, appliqué ici à chaque étage pour observer l'évolution. Cette évolution est positive, en effet plus notre architecture de cascade contient d'étages plus les performances avec décodage dirigé par le lexique s'améliorent. Chaque réseau apporte donc de l'information supplémentaire même avec la contrainte du décodage dirigé par le lexique. On observe une diminution du taux d'erreur de 19% entre une cascade à un réseau et une cascade à 3 réseaux.
Nous cherchons désormais à savoir si cette première conclusion se confirme sur la tâche WR3 et à observer l'effet de l'augmentation de la taille du lexique sur les performances de notre cascade.

Réseau	**Lecture**	**Erreur**	**Rejet**	**MC**
1er étage	66.37	2.86	30.78	4.1
2ème étage	73.84	3.38	22.78	6.5
3ème étage	77.76	3.76	18.48	8.8

TABLE 4 – Résultats de la cascade pour la tâche WR3

Réseau	**Lecture**	**Erreur**	**Rejet**
1er étage	88.72	11.28	0
2ème étage	89.99	10.01	0
3ème étage	90.63	9.37	0

TABLE 5 – Résultats de la cascade pour la tâche WR3 avec décodage dirigé par le lexique en sortie de chaque étage

Les tableaux 4 et 5 contiennent les résultats de la même expérimentation pour la tâche WR3. Le premier élément remarquable est que les taux de lecture sont identiques à l'issue de la première cascade par rapport à WR2, seul le taux d'erreur est très légèrement plus élevé. Cette augmentation du taux d'erreur est due à l'augmentation de la taille du lexique de 1692 à 5744 mots, engendrant des confusions. Sur les critères précédents, nous retrouvons des résultats très similaires avec une classification de 26% et 18.9% des rejets à l'issu des étages 2 et 3 respectivement. Cependant nous avions remarqué une augmentation de la mauvaise classification, celle-ci se retrouve dans les taux d'erreur des rejets classés par les réseaux 2 et 3 qui sont respectivement de 6.5% et 8.8%. Nous tirons les mêmes conclusions avec le décodage dirigé par le lexique, ou l'erreur diminue au cours des niveaux de la cascade, avec une diminution de l'erreur de 17% entre le premier et le dernier niveau de la cascade.
On observe entre les résultats de WR2 et de WR3 que le taux de lecture ne varie quasiment pas de 78.07 à 77.76 soit une différence de 0.31 points, grâce à la vérification lexicale. Le dictionnaire de la tâche WR2 étant inclut dans celui de la tâche WR3, un mot correctement vérifié par le lexique en WR2 l'est également en WR3. La légère différence est due au fait qu'il y a un peu plus d'erreurs à cause de l'augmentation de la taille du lexique, certaines de ces erreurs étaient rejetées et corrigées dans un étage supérieur en WR2. Il faut cependant noter que si la baisse du taux de lecture est très limitée, le taux d'erreur évolue un peu plus significativement (0.94 points).
Nous nous comparons maintenant au système de TUM ayant obtenu les meilleurs résultats de la compétition RIMES 2009 (n'ayant pas de résultat pour WR2 en 2011), toutes catégories confondues. Ce système est celui dont la sensibilité à la taille du lexique est la plus faible de la compétition. TUM a une différence de lecture entre WR2(93.2%, 1612 mots) et WR3(91%, 4943 mots) de 2.36% alors que notre système présente, entre la tâche WR2(1692 mots) et WR3(5744 mots), une différence considérablement plus faible, de seulement 0.39%.
Grâce à cette différence de lecture très faible entre WR2 et WR3, nous pouvons déduire que notre méthode est très peu sensible à la taille de lexique. Notre système a également la particularité d'avoir un taux d'erreur très faible de 3.76%, nous plaçant premier sur l'erreur par rapport aux systèmes de la compétition RIMES 2011 [12]. Cependant notre taux de mots biens reconnus est bien moins bons à cause des rejets et c'est en quatrième position que nous serions. Afin d'obtenir une meilleure comparaison, nous appliquons un décodage dirigé par le lexique sur les rejets finaux de cascade, nous obtenons un taux d'erreur très encourageants de 9.37%, nous plaçant deuxième par rapport aux systèmes de la compétition RIMES 2011, en sachant que le premier utilise une combinaison de 7 classifieurs dont 4 réseaux LSTM avec HMM.

5 Conclusion

Dans cet article nous avons proposé une architecture de cascade de BLSTM avec un système décisionnel par vérification lexicale. Cette nouvelle méthode possède des pro-

priétés intéressantes qui sont la faible sensibilité au lexique, un taux d'erreur très faible, une rapidité face au grand lexique et une capacité de rejet. En traitant les rejets finaux à l'aide d'un décodage dirigé par le lexique, nous obtenons des résultats prometteurs proches des meilleures méthodes actuellement à l'état de l'art. Un des points faibles de notre cascade est l'impossibilité de rattraper une erreur qui a été commise à un étage précédent. Un autre point faible est que nous ne pouvons pas faire de reconnaissance hors lexique, du fait de sa nécessité pour la vérification. Les perspectives de ces travaux portent sur la réalisation d'un meilleur contrôle de l'erreur via notre système de décision, ainsi que l'étude d'autres réseaux afin d'améliorer leurs complémentarités.

Références

[1] G. Bideault, L. Mioulet, C. Chatelain, and T. Paquet. Spotting handwritten words and regex using a two stage blstm-hmm architecture. In *Document Recognition and Retrieval, San Francisco, USA*, 2015.

[2] Anja Brakensiek, Jörg Rottland, and Gerhard Rigoll. Handwritten address recognition with open vocabulary using character n-grams. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 357–362. IEEE, 2002.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[4] Haikal El Abed, Volker Margner, Monji Kherallah, and Adel M Alimi. Icdar 2009 online arabic handwriting recognition competition. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1388–1392. IEEE, 2009.

[5] A El-Yacoubi, Michel Gilloux, Robert Sabourin, and Ching Y. Suen. An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8) :752–760, 1999.

[6] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 369–376, 2006.

[7] Alex Graves. Rnnlib : A recurrent neural network library for sequence learning problems. `https://sourceforge.net/projects/rnnl`.

[8] Alex Graves. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

[9] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5) :602–610, 2005.

[10] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 545–552, 2009.

[11] Emmanuele Grosicki and Haikal El Abed. Icdar 2009 handwriting recognition competition. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 1398–1402. IEEE, 2009.

[12] Emmanuele Grosicki and Haikal El-Abed. Icdar 2011-french handwriting recognition competition. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1459–1463. IEEE, 2011.

[13] Mahdi Hamdani, Amr El-Desoky Mousa, and Hermann Ney. Open vocabulary arabic handwriting recognition using morphological decomposition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 280–284. IEEE, 2013.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[15] Alessandro L Koerich, Robert Sabourin, and Ching Y Suen. Large vocabulary off-line handwriting recognition : A survey. *Pattern Analysis & Applications*, 6(2) :97–121, 2003.

[16] Farès Menasri, Jérôme Louradour, Anne-Laure Bianne-Bernard, and Christopher Kermorvant. The a2ia french handwriting recognition system at the rimes-icdar2011 competition. In *IS&T/SPIE Electronic Imaging*, pages 82970Y–82970Y. International Society for Optics and Photonics, 2012.

[17] L. Mioulet, G. Bideault, C. Chatelain, T. Paquet, and S. Brunessaux. Exploring multiple feature combination strategies with a recurrent neural network architecture for off-line handwriting recognition. In *Document Recognition and Retrieval, San Francisco, USA*, pages 94020F–94020F, 2015.

[18] Réjean Plamondon and Sargur N Srihari. Online and off-line handwriting recognition : a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1) :63–84, 2000.

[19] Thomas Plötz and Gernot A Fink. Markov models for offline handwriting recognition : a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(4) :269–298, 2009.

[20] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.

[21] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11) :2673–2681, 1997.

[22] Andrew Senior and Tony Robinson. Forward-backward retraining of recurrent neural networks. *Advances in Neural Information Processing Systems*, pages 743–749, 1996.

[23] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[24] Andrew J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2) :260–269, 1967.

[25] Bailing Zhang. Reliable classification of vehicle types based on cascade classifier ensembles. *Intelligent Transportation Systems, IEEE Transactions on*, 14(1) :322–332, 2013.

[26] Ping Zhang, Tien D Bui, and Ching Y Suen. A novel cascade ensemble classifier system with a high recognition performance on handwritten digits. *Pattern Recognition*, 40(12) :3415–3429, 2007.