Détection d'erreurs dans des transcriptions OCR de documents historiques par réseaux de neurones récurrents multi-niveau

Thibault Magallon Frederic Bechet Benoit Favre
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
prenom.nom@lis-lab.fr

RÉSUMÉ
Le traitement à posteriori de transcriptions OCR cherche à détecter les erreurs dans les sorties d'OCR pour tenter de les corriger, deux tâches évaluées par la compétition ICDAR-2017 Post-OCR Text Correction. Nous présenterons dans ce papier un système de détection d'erreurs basé sur un modèle à réseaux récurrents combinant une analyse du texte au niveau des mots et des caractères en deux temps. Ce système a été classé second dans trois catégories évaluées parmi 11 candidats lors de la compétition.

ABSTRACT
Combining character level and word level RNNs for post-OCR error detection

Post-OCR processing, consist in detecting errors first, then correcting them when possible. In this context the ICDAR-2017 Competition on Post-OCR Text Correction was organized to compare approaches on these two tasks. This paper presents an OCR error detection system based on a 2-pass RNN model combining character level and word level representations. This system was ranked 2nd on three datasets among 11 participants at the ICDAR-2017 Competition.

MOTS-CLÉS : OCR, detection d'erreurs, réseaux de neurones récurrents.

KEYWORDS: OCR, error detection, recurrent neural networks.

1 Introduction

Les systèmes actuels de Reconnaissance Optique de Caractères (Optical Character Recognition - OCR) obtiennent désormais d'excellentes performances sur des documents imprimés et scannés avec soin. Cependant les documents historiques restent un défi pour les domaines du traitement d'images et du traitement automatique de la langue du fait d'une mauvaise qualité d'impression ainsi que de supports parfois endommagés. De plus, certaines collections de documents numérisés à l'aide de ces outils ne sont que rarement retraitées avec des systèmes à jours, principalement pour des raisons de coûts. Or ces erreurs d'OCR peuvent avoir un impact conséquent sur la recherche de documents dans une bibliothèque numérique (Chiron *et al.*, 2017b). Ainsi, indépendamment de la tâche d'OCR elle-même, le post-traitement des transcriptions automatiques est une tâche permettant à la fois d'évaluer la qualité d'archives numérisées tout en donnant l'occasion de les corriger. C'est dans ce contexte que la compétition *ICDAR-2017 Post-OCR Text Correction* a été organisée, dans le but de comparer différentes approches concernant les deux tâches de détection d'erreurs d'une part et de correction des erreurs détectées d'autre part. En tenant compte des contraintes liées à ces archives,

seule la sortie de texte brute est accessible, aucune autre information n'est fournie, comme l'image, les scores de confiances ou les informations relatives aux polices détectées.

Historiquement, ces systèmes utilisent des modèles de caractères ainsi que des collections de mots pour résoudre certaines ambiguïtés survenant après l'analyse d'image (Bokser, 1992). Mais il arrive que des erreurs subsistent à l'utilisation de telles méthodes, car le vocabulaire ne peut être couvert pour toute la langue d'une part et pour des questions propres au domaine de l'OCR ou l'utilisation de modèles reposant uniquement sur la fréquence des mots peuvent s'avérer insuffisants (Smith, 2011). De nombreuses méthodes ont été proposées pour simultanément détecter et corriger ces erreurs, en utilisant différentes approches tels que les canaux bruités (Kolak & Resnik, 2002; Evershed & Fitch, 2014), l'utilisation d'outils externes tels que des correcteurs orthographiques (Bassil & Alwani, 2012; Schulz & Kuhn, 2017) ou la combinaison de plusieurs systèmes d'OCR afin d'accroître la robustesse de la reconnaissance (Abdulkader & Casey, 2009). De plus, il n'est pas rare dans le cas de la correction d'erreurs à posteriori d'étiqueter les corrections à apporter sur une séquence donnée afin de la remanier par la suite, comme cela peut être fait pour la post-édition automatique de traductions (Libovickỳ *et al.*, 2016; Bérard *et al.*, 2017).

Le présent document décrit l'architecture proposée par notre équipe pour la *tâche de détection d'erreurs* de la compétition ICDAR. Nous avons suivi le même type d'approche ayant été appliquée avec succès aux méthodes de détection d'erreurs dans des sorties de Reconnaissance Automatique de la Parole (Béchet & Favre, 2013), en considérant la tâche de détection d'erreur comme un exercice de classification de séquences. La méthode repose sur un modèle de réseaux de neurones récurrent analysant le texte à l'échelle des caractères ainsi qu'à celle des mots, le tout joint dans un seul modèle multilingue s'exécutant en deux temps. Ce système a été utilisé sur un corpus rassemblé pour l'occasion contenant des documents historiques en anglais et français et a été classé second dans le cadre de la tâche de détection parmi onze participants.

2 La compétition Post-OCR Text Correction

La compétition ICDAR-2017 Post-OCR (Chiron *et al.*, 2017a) a été séparée en deux tâches afin d'évaluer les différentes méthodes proposées par chaque compétiteur pour la détection ainsi que la correction d'erreurs dans des textes produits par un système OCR.

Le corpus fourni dans ce cadre est composé d'extraits de monographies et de périodiques rédigés en français ou en anglais provenant d'archives de la Bibliothèque Nationale Française (BnF) et de la British Library (BL). Ce jeu de données regroupe au total plus de douze millions de caractères dont les textes sont issus d'une période couvrant les quatre siècles derniers. L'ensemble de ces textes provient d'une sous-partie d'un corpus collecté dans le cadre du projet AmeliOCR, mené par le laboratoire L3i (Université de La Rochelle, France) et la Bibliothèque nationale Française. Ces deux tâches de post-traitement sont difficiles, car si les technologies d'OCR ont aujourd'hui acquis une certaine maturité, la qualité des supports source et la structure des documents ainsi que le vocabulaire ancien et varié ajoute une autre difficulté à l'extraction du texte contenu dans les images numérisées, produisant des sorties textuelles erronées nécessitant une correction.

Afin de pouvoir évaluer les détections et corrections apportées, des annotations *Gold-Standard* (GS) accompagnent les sorties de texte brut. Ces dernières ont étés produites manuellement et sont alignées au niveau des caractères.

La distribution des sources dans ce corpus de 12M de caractères en fonction de la langue et du type de document est référencée dans le Tableau 1. On observe ainsi que le taux d'erreur OCR peut varier de 1% à 4%.

English				
corpus	*type*	*période*	*%erreur*	*taille*
BL Euro NP	périodique	1744 - 1894	4%	1.8 M
BL Monog	monographie	1858 - 1891	1%	1.2 M
GT BnF Eng	monographie	1802 - 1911	2%	3.0 M
French				
corpus	*type*	*période*	*%erreur*	*taille*
Europeana NP	périodique	1814 - 1944	4%	1.0 M
IMPACT	monographie	1821 - 1864	1%	0.4 M
GT BnF Fr	mélangé	1686 - 1943	1%	2.0 M
Digit. BnF	mélangé	1654 - 2000	3%	0.2 M
News other	périodique	1897 - 1934	4%	0.6 M
Monog other	monographie	1689 - 1883	3%	1.8 M

TABLE 1: Distribution du corpus ICDAR Post-OCR.

3 Réseau de neurones récurrent au niveau des mots et des caractères pour la détection d'erreurs

La première étape de notre système consiste en une combinaison de deux modèles de réseaux de neurones récurrents (*Recurrent Neural Network*, abrégé *RNN* par la suite) au niveau des caractères pour associer une étiquette (*correct* ou *erroné*) à chaque symbole d'une séquence, ainsi que d'un modèle de langue dont le rôle est de prédire le caractère suivant étant donné les caractères l'entourant. Ces deux éléments sont présentés dans les sous-sections suivantes. L'implémentation de ce réseau a été réalisée à l'aide du toolkit Keras (Chollet *et al.*, 2015).

3.1 Classification binaire à l'échelle des caractères

Ce premier RNN est une implémentation directe de la tâche de détection d'erreurs au niveau des caractères. C'est-à-dire qu'étant donné un certain contexte, il doit prédire si l'apparition d'un symbole est erroné ou non, sans aucune autre information d'entrée que la séquence textuelle.

Concernant l'entrée du modèle, nous l'avons fixée à un vecteur de dimension 64, où pour tout élément de celui-ci est associé un symbole issu du vocabulaire. Chaque chaîne du texte est alors découpé en plusieurs séquences de ce format (si cela est nécessaire). De plus, une représentation vectorielle de cette entrée est utilisé (plongement de mots) à l'aide d'une fenêtre regroupant les quatorze voisins d'un terme cible.

La sortie de cette couche nous permet d'obtenir une représentation vectorielle de notre séquence d'entrée. Elle est par la suite dirigée vers une couche récurrente afin de considérer les erreurs présente dans celle-ci et ainsi effectuer la tâche de classification associée à la détection d'erreurs. Cette couche récurrente implémente un modèle à mémoire de type *Gated Recurrent Units* (GRU) (Cho *et al.*, 2014).

Nous utilisons 64 neurones pour la couche récurrente, soit autant que de symboles qui composent nos vecteurs d'entrée. De plus, nous ajoutons à cette couche des propriétés de bidirectionnalité ainsi que le fait de ne pas réinitialiser l'état caché pour chaque nouveau vecteur d'entrée (modèle *stateful*). Pour finir, une couche dense de deux unités sur laquelle nous appliquons une activation de type *softmax* est utilisée afin de prédire pour chaque symbole les probabilités d'appartenance aux deux classes, *correct* et *incorrect*.

3.2 Le modèle de langue à l'échelle des caractères

Le modèle récurrent de classification binaire décrit précédemment pourrait-être utilisé tel quel pour la tache de prédiction d'erreurs, cependant ce dernier souffre d'un défaut non-négligeable : il ne peut qu'être entraîné sur un corpus de sorties d'OCR accompagné des erreurs annotées. Cette source de données est rare, même dans le contexte de la compétition et sa quantité est donc restreinte. C'est pourquoi, nous avons utilisé des données complémentaires afin d'améliorer notre méthode de classification, en tirant profit d'une grande quantité de texte provenant d'un corpus de textes journalistiques, nous permettant ainsi d'établir et d'apprendre des statistiques propres aux langues sur lesquelles nous souhaitons détecter les erreurs. Néanmoins, nous ne pouvons directement ajouter de telles données au corpus d'entraînement puisqu'aucun symbole de ce dernier n'est annoté pour des erreurs d'OCR. Mais nous pouvons les ajouter en tant que *Modèle de Langue* (ML), à l'échelle des caractères, afin de prédire le symbole suivant étant donné la position d'un caractère dans une séquence et compte tenu du contexte l'entourant. L'idée derrière l'ajout d'un tel mécanisme est de fournir au système de classification binaire une information supplémentaire quant à la régularité d'apparition de certains termes au travers de la représentation du ML. En effet, un enchaînement peu probable dans la séquence peut modifier le degré de confiance du modèle s'il ne possède qu'une faible probabilité d'apparition compte tenu de sa place dans la séquence et des schémas observés lors de l'apprentissage.

L'entrée du modèle de langue se fait de façon similaire à celle utilisée pour la classification binaire que nous avons décrite précédemment. Concernant sa structure, le réseaux est en tout constitué de trois couches. Une de plongement de mots, une récurrente et une dense d'activation, possédant un nombre d'unités égale au nombre d'éléments présent dans le vocabulaire des symboles.

Cette couche d'activation est par la suite concaténée à la sortie de la représentation vectorielle des entrées du modèle de classification binaire.

3.3 Le modèle récurrent à l'échelle des mots

En complément du traitement fait pour les caractères, nous avons ajouté un paradigme similaire opérant au niveau des mots afin de pouvoir prendre en compte des dépendances plus lointaines et des contraintes plus fortes, notamment syntaxiques, dans notre système de détection d'erreurs. Ce modèle se voit donc doté d'une structure similaire à celle établie pour les symboles. La principale différence résidant dans le fait que celui-ci doit inclure les informations déduites au niveau des caractères dans son traitement de détection d'erreurs sur les mots. Pour cela, nous ajoutons à cette partie du système une entrée sous la forme d'une représentation issue de la couche d'activation de la détection de symboles erronés. Ce système que nous avons utilisé lors de la compétition peut être schématisé par la Figure 1.

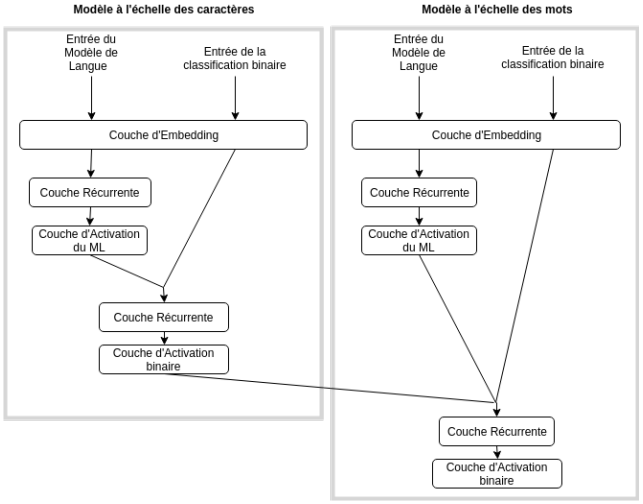

FIGURE 1: Schématisation du système combinant la représentation à l'échelle des caractères et des mots.

4 Configuration expérimentale

Pour entraîner le système ayant participé à la compétition ICDAR Nous avons utilisé $\frac{2}{3}$ de la totalité du corpus que nous avions à disposition (tableau 1) comme corpus d'entraînement et les $\frac{1}{3}$ restant comme corpus de validation. Les modèles de langue à l'échelle des caractères mais aussi à celle des mots ont étaient entraînés sur la portion du corpus de la compétition réservée à cet effet, ainsi que sur des textes issus de dépêches AFP des années passées afin d'accroître le vocabulaire et de permettre au ML de pouvoir effectuer de meilleures généralisations. Nous avons ajouté ces données à hauteur de 25 000 nouveaux termes pour chaque langage (français et anglais), ceci semblant être le seuil optimal pour lequel l'ajout de ces éléments de lexique puisse apporter des gains, avant d'engendrer une dégradation des performances sur le corpus d'évaluation. Notre modèle utilise dans sa version finale un lexique de 256 symboles caractères et de 136 752 mots, incluant la ponctuation.

Il est important de noter que la détection d'erreurs est un problème très déséquilibré au niveau de la représentation des données. En effet, ces dernières représentent, dans le corpus mis à notre disposition durant la compétition, approximativement 2% de la totalité des mots. Un réseau de neurones entraîné sur une telle collection aura tôt fait de toujours choisir comme résultat de prédiction la catégorie d'exemples dominante, c'est-à-dire la classe *non-erronée*. Pour surmonter cette difficulté, nous avons, durant la post-propagation ayant lieu lors de l'apprentissage, doublé le poids attribué par la fonction de coût lorsqu'une mauvaise prédiction devant être attribuée à la classe d'erreur se présentait. Cela dans le but d'augmenter l'impact des corrections faites au modèle durant cette phase.

5 Résultats et évaluation

La tache de détection d'erreurs de la compétition Post OCR Text Correction est évaluée à l'échelle des tokens, qui ne sont autres, pour les organisateurs, qu'une suite de symboles séparés par un espacement

(incluant les tabulations et les caractères vides), ponctuation comprise. Les métriques utilisées sont le *Rappel*, la *Précision* et la *F-mesure*, avec un classement des participants effectué sur la *F-mesure*.

Le jeu de données de test contient 177K mots (96,5K en anglais et 80,6K en français) avec des taux d'OCR bien plus élevés que pour les corpus d'entraînement (de 7 à 10%).

Comme on peut le voir dans le tableau 2, les résultats officiels de la compétition ICDAR (Chiron *et al.*, 2017a) placent notre système à la seconde position dans trois des quatre catégories de textes. Les résultats obtenus sont stables pour la langue anglaise indépendamment du type de document et suivent la même chute de performances lorsque nous comparons les scores de nos sorties à celles des gagnants de cette compétition sur la partie du corpus attribuée aux textes français, en particuliers les monographies comme décrit dans le Tableau 2. Ceci peut être en partie expliqué par le fait que le taux d'erreur est moindre sur le corpus de textes français, les erreurs étant dès lors plus difficiles à mettre en évidence.

	Mono-EN	**Perio-EN**	**Mono-FR**	**Perio-FR**	**Confondus**
A	0.73	0.68	0.55	0.69	0.66
X	**0.66**	**0.66**	**0.43**	**0.60**	0.58
B	0.69	0.54	0.40	0.54	0.54
C	0.67	0.64	0.31	0.50	0.53
D	0.66	0.44	0.36	0.41	0.46

TABLE 2: Résultats officiels pour la tâche de détection en F-mesure. Notre système y est noté **X**.

Nous avons généré des résultats comparatifs durant la période de la compétition afin de valider notre approche en deux étapes sur le corpus de développement. Ces résultats sont regroupés dans le Tableau 3. **C-RNN** fait référence au système de classification binaire à l'échelle des symboles, **C-ML** le Modèle de Langue à l'échelle des caractères, **M-RNN** et **M-ML** désignent les mêmes types de systèmes, à l'échelle des mots. Comme nous pouvons le voir, le modèle **C-RNN** obtient la meilleure précision, mais un rappel très faible lorsqu'il est entraîné seulement sur les données du corpus de la compétition. L'ajout du ML accompagné de données additionnelles provenant des dépêches AFP augmente fortement le rappel. Les modèles à l'échelle des mots, sont quant à eux plus robustes. Cependant les combiner au travers d'un unique modèle permet un gain global des performances.

Modèles	**F-mesure**	**Rappel**	**Précision**
C-RNN	0.24	0.14	**0.62**
C-RNN + C-ML	0.45	0.42	0.48
M-RNN + M-ML	0.53	0.65	0.45
Modèle final	**0.55**	**0.67**	0.46

TABLE 3: Résultats comparatifs dépendemment du modèle utilisé

6 Conclusion

Nous avons présenté dans ce papier un système de détection d'erreurs pour des textes issus de sorties d'OCR dans les langues anglaise et française, développé pour la compétition Post-OCR Text Correction ayant eu lieu lors de la conférence ICDAR-2017. Afin de détecter les possibles erreurs de

telles sorties, nous avons proposé une approche basée sur l'analyse à différents niveau d'informations textuelles dans un réseau de neurones récurrent. La première partie de ce système est entraînée à l'échelle des caractères et la seconde à celle des mots accompagnés d'informations alignées sur chaque mot provenant du modèle à l'échelle des symboles. Cette architecture de classification pour la détection d'erreurs a obtenu de bons résultats lors de l'évaluation de la compétition ICDAR 2017 et notre système a été classé second parmi 11 participants.

Références

ABDULKADER A. & CASEY M. R. (2009). Low cost correction of ocr errors using learning in a multi-engine environment. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, p. 576–580 : IEEE.

BASSIL Y. & ALWANI M. (2012). Ocr post-processing error correction algorithm using google online spelling suggestion. *arXiv preprint arXiv :1204.0191*.

BÉCHET F. & FAVRE B. (2013). Asr error segment localization for spoken recovery strategy. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, p. 6837–6841 : IEEE.

BÉRARD A., PIETQUIN O. & BESACIER L. (2017). Lig-cristal system for the wmt17 automatic post-editing task. *arXiv preprint arXiv :1707.05118*.

BOKSER M. (1992). Omnidocument technologies. *Proceedings of the IEEE*, **80**(7), 1066–1078.

CHIRON G., DOUCET A., COUSTATY M., VISANI M. & MOREUX J.-P. (2017a). ICDAR2017 competition on post-OCR text correction. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR2017)* : IEEE.

CHIRON G., DOUCET A., COUSTATY M., VISANI M. & MOREUX J.-P. (2017b). Impact of ocr errors on the use of digital libraries : Towards a better access to information. In *Digital Libraries (JCDL), 2017 ACM/IEEE Joint Conference on*, p. 1–4 : IEEE.

CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*.

CHOLLET F. *et al.* (2015). Keras. https ://github.com/fchollet/keras.

EVERSHED J. & FITCH K. (2014). Correcting noisy ocr : Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, p. 45–51 : ACM.

KOLAK O. & RESNIK P. (2002). Ocr error correction using a noisy channel model. In *Proceedings of the second international conference on Human Language Technology Research*, p. 257–262 : Morgan Kaufmann Publishers Inc.

LIBOVICKÝ J., HELCL J., TLUSTÝ M., PECINA P. & BOJAR O. (2016). Cuni system for wmt16 automatic post-editing and multimodal translation tasks. *arXiv preprint arXiv :1606.07481*.

SCHULZ S. & KUHN J. (2017). Multi-modular domain-tailored ocr post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 2706–2716.

SMITH R. (2011). Limits on the application of frequency-based language models to ocr. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, p. 538–542 : IEEE.