

# Exploiter des modèles de langue pour évaluer des sorties de logiciels d'OCR pour des documents français du XVII<sup>e</sup> siècle

Jean-Baptiste Tanguy

CELLF, STIH, Sorbonne Université, 1 rue Victor Cousin, 75005, Paris, France  
jean-baptiste.tanguy@sorbonne-universite.fr

## RÉSUMÉ

---

Pour comparer deux sorties de logiciels d'OCR, le *Character Error Rate* (ou, *CER*) est fréquemment utilisé. Moyennant l'existence d'une vérité de terrain de qualité pour certains documents du corpus, le *CER* calcule le taux d'erreur de ces pièces et permet ensuite de sélectionner le logiciel d'OCR le plus adapté. Toutefois, ces vérités de terrain sont très coûteuses à produire et peuvent freiner certaines études, même prospectives. Nous explorons l'exploitation des modèles de langue en agrégeant selon différentes méthodes les probabilités offertes par ceux-ci pour estimer la qualité d'une sortie d'OCR. L'indice de corrélation de Pearson est ici utilisé pour comprendre dans quelle mesure ces estimations issues de modèles de langue covarient avec le *CER*, mesure de référence.

## ABSTRACT

---

### Language Model Based Evaluation of OCR Software Output Qualities for 17th Century French Documents

In order to compare two OCR software outputs, the *Character Error Rate* (or, *CER*) is frequently used. When a quality ground truth exists for several documents, the *CER* calculates the error rate for these documents and therefore allows to choose the most suitable OCR software. However, these ground truths are extremely expensive and may slow down some studies. Hence, we are exploring the exploitation of language models by the aggregation (with several methods) of their probabilities in order to estimate OCR output qualities. The Pearson correlation is used to understand how these language model based estimations covary with the *CER*, reference metric.

---

**MOTS-CLÉS** : OCR, modèle de langue, évaluation, document historique, français pré-classique.

**KEYWORDS** : OCR, language model, evaluation, historical document, pre-classical French.

---

## 1 Introduction

Les campagnes de numérisation des collections patrimoniales s'installent à la frontière de deux enjeux relatifs aux documents historiques : leur pérennisation et leur accessibilité. La Bibliothèque Nationale de France, qui a commencé la numérisation de ses fonds au début des années 1990 (avec l'arrivée de Gallica en 1997 (Bermes, 2020)), et la Bibliothèque Mazarine, qui a engagé en 2014 la numérisation de sa collection d'incunables et en 2015 celle de ses Mazarinades<sup>1</sup>, sont ici exemplaires. Au-delà de la construction d'éditions web, la numérisation de telles collections rend possible leur exploitation automatique à grande échelle, moyennant une transcription de leur contenu textuel. Ceci

---

1. Documents parus en France, lors de la Fronde (1648-1653).

constitue un réel intérêt, tant pour la communauté savante que pour le grand public. Toutefois, deux problèmes majeurs se posent. D'une part, les logiciels de reconnaissance optique de caractères (ou OCR), s'ils offrent des transcriptions automatiques de qualité pour des documents contemporains générés électroniquement, sont nettement moins robustes face à des documents historiques. (Lejeune & Abiven, 2019), pour le « corpus » des Mazarinades, exposent un ensemble d'éléments rendant l'étude de ces documents historiques particulièrement complexe : variantes graphiques, abréviations, orthographe erratique mais aussi un « état inégal de conservation [des] imprimés souvent produits dans l'urgence et l'économie de moyens (papier et encre de mauvaise qualité, notamment) ». D'autre part, et s'agissant de processus automatisés, la connaissance de la qualité des sorties des logiciels d'OCR est primordiale. Néanmoins, l'évaluation d'outils d'OCR n'est pas stable d'un corpus à l'autre, car elle fait intervenir des corpus particulièrement hétérogènes (Springmann *et al.*, 2014). Mesurer la qualité d'une sortie d'OCR nécessite alors, au moins pour un ensemble réduit de la collection à numériser et à océriser, une transcription manuelle et certaine à laquelle les sorties d'OCR seront comparées ; et ce, dès lors qu'une nouvelle collection est à océriser. Or cette transcription de référence, qu'on appelle vérité de terrain (Springmann *et al.*, 2018), est coûteuse à constituer ce qui limite d'autant la quantité de données disponible pour l'évaluation. Ainsi, estimer la qualité des sorties de logiciels d'OCR sans vérité de terrain permettrait d'opter à moindre coût pour un logiciel d'OCR adapté. Il s'agit donc d'une démarche d'évaluation non supervisée.

Dans cet article, nous proposons i) d'apprendre des modèles de langue sur un corpus en français pré-classique (XVII<sup>e</sup> siècle), ii) de parcourir des sorties de logiciels d'OCR par fenêtre glissante en récupérant les probabilités de chaque modèle de langue de rencontrer une telle séquence de caractères pour enfin iii) estimer la qualité de ces sorties d'OCR. Différentes méthodes d'agrégation des probabilités précitées sont proposées pour estimer la qualité globale des *pages* océrisées. L'étude des corrélations entre ces estimations et les *CER*<sup>2</sup> (pour chaque page du corpus) permettra de valider ou réfuter la pertinence de ces estimateurs sur le corpus de l'étude.

Après l'exposition de plusieurs méthodologies d'estimation non supervisée de la qualité de sorties d'OCR (section 2), nous décrivons le cadre expérimental de notre étude, où le corpus, les modèles de langue et les méthodes d'agrégation de leurs probabilités sont décrits (section 3). Nous présenterons finalement les résultats de l'expérimentation en section 4.

## 2 Estimer la qualité de sorties de logiciels d'OCR

Pour évaluer des sorties d'OCR en échappant à la fastidieuse transcription des imprimés, plusieurs méthodes conduisent à la création de nouvelles mesures d'évaluation. Celles-ci sont comparées à des mesures de référence (le *CER* ou la précision), calculées grâce à des vérités de terrain, pour valider ou réfuter leur pertinence.

**Exploiter des ressources lexicales** (Springmann *et al.*, 2016) proposent d'estimer la qualité d'une sortie d'OCR en exploitant la *lexicalité* de celle-ci. La *lexicalité* est calculée en faisant la moyenne, pour chaque *token* observé dans la sortie d'OCR, des distances de Levenshtein entre ces *tokens* et leur supposé équivalent moderne le plus proche (*supposé* car la relation entre deux formes de deux états différents d'une même langue n'est pas nécessairement bijective). Il est montré que cette *lexicalité*

---

2. *Character Error Rate*.

est très clairement corrélée à la précision.

**Exploiter les valeurs de confiance des logiciels d'OCR** Une autre voie empruntée par (Springmann *et al.*, 2016) est de mettre à profit les valeurs de confiance des logiciels d'OCR. Ceux-ci renvoient en effet une valeur correspondant à l'intensité de la confiance que le logiciel associe au caractère qu'il propose. Dans le cas d'une hésitation entre deux caractères proches (par exemple, *G* et *O*), le conflit est traduit par deux valeurs de confiance similaires et plus faibles que dans le cas d'une certitude pour un caractère en particulier. Les auteurs supposent que « la somme des valeurs de confiance associées aux caractères de sortie doit ainsi être corrélée avec la précision de la sortie d'OCR »<sup>3</sup> et le vérifient très nettement.

**Exploiter les *bounding boxes*** Avant de proposer un ensemble de caractères, les logiciels d'OCR segmentent les images proposées à l'océrisation. Ces segmentations (en colonnes, en lignes, en mots ou encore en caractères) apparaissent pour (Gupta *et al.*, 2015) comme de bons indicateurs pour estimer la qualité d'une sortie d'OCR. En effet, s'agissant d'un processus en cascade, si la segmentation fait défaut, l'océrisation en pâtira largement. En recueillant les informations graphiques associées aux objets résultant de la segmentation (les *bounding boxes*), les auteurs proposent d'apprendre un modèle de classification permettant de distinguer deux types de *bounding boxes* : les *bounding boxes* pertinentes (*BBs*) et les *bounding boxes* non pertinentes (*noise BBs*)<sup>4</sup>. Le calcul de cet estimateur se réalise ensuite en comptant la proportion des *noise BBs*. Il est conclu que cette proportion de *noise BBs* permet d'estimer de manière satisfaisante la qualité globale d'un document océrisé mais aussi d'en identifier les passages bruités.

**Exploiter les modèles de langue** Les modèles de langue, appris au grain mot, sont fréquemment utilisés en reconnaissance de la parole. (Chen *et al.*, 1998) ont proposé d'utiliser les modèles de langue non pas pour corriger en post-traitement les sorties de reconnaissance d'un flux de parole mais pour estimer la qualité de cette sortie. La perplexité et ses dérivés (comme la log-perplexité) y apparaissent fortement corrélées au *word error rate* (le taux d'erreur mot, ou, *WER*) avec, pour le premier jeu de données de leur étude, une relation presque parfaitement linéaire. Néanmoins, pour l'évaluation de sorties d'OCR, les modèles de langue ne semblent pas avoir encore été testés.

**Utiliser des pseudo-vérités de terrain** (Ul-Hasan *et al.*, 2016) proposent d'utiliser la sortie d'un logiciel d'OCR (en l'occurrence, Tesseract) comme une pseudo-vérité de terrain sur laquelle est appris un premier modèle. Si l'objectif de ce travail n'est pas d'estimer la qualité d'une sortie d'OCR, les auteurs se soucient du manque de transcriptions à disposition et atteignent avec ces pseudo-vérités de terrain des précisions de l'ordre de 95% sur des documents imprimés du XVII<sup>e</sup> siècle.

Comme (Chen *et al.*, 1998), nous proposons d'utiliser des modèles de langue (mais appris au grain caractère) pour évaluer les sorties des logiciels d'OCR. Il s'agit de récupérer une probabilité pour chaque caractère (voir sous-section 3.4), d'agrèger ces probabilités et d'observer, comme (Springmann *et al.*, 2016), s'il existe une corrélation entre ces agrégations et la mesure de référence *CER*.

---

3. *The sum of the confidences over all output characters should therefore correlate with the accuracy of the output.*

4. Par exemple, une *bounding box* pertinente encadre une ligne ou un mot alors qu'une *bounding box* non pertinente encadre deux lignes juxtaposées mais appartenant à deux colonnes différentes.

### 3 Cadre expérimental

Dans cette section, nous décrivons le corpus de notre étude, la méthodologie et les logiciels d’océri- sation, les modèles de langue appris et les mesures d’évaluation à comparer au *CER*.

#### 3.1 Un corpus d’œuvres françaises du XVII<sup>e</sup> siècle

Titre	Auteur	Date	Domaine	Nb pages	Nb lignes	Nb mots
<i>Oraisons funebres</i>	Bossuet	1683	Théologie	27	770	4 128
<i>La Pucelle...</i>	Chapelain	1656	Poésie	28	753	4 735
<i>Advis sur la peste</i>	Ellain	1606	Science	22	618	3 168
<i>Egalite des hommes et des femmes</i>	Gournay	1622	Philosophie	31	825	4 284
<i>La Maniere d’amolir les os...</i>	Papin	1682	Science	23	548	2 230
<i>Experiences Nouvelles...</i>	Pascal	1647	Science	39	776	3 568
<i>Introduction à la vie devote</i>	Sales	1641	Théologie	25	618	3 915
<i>Oeuvres completes (Tome II.)</i>	Viau	1623	Poésie	33	852	4 055

TABLE 1 – Description des œuvres du corpus.

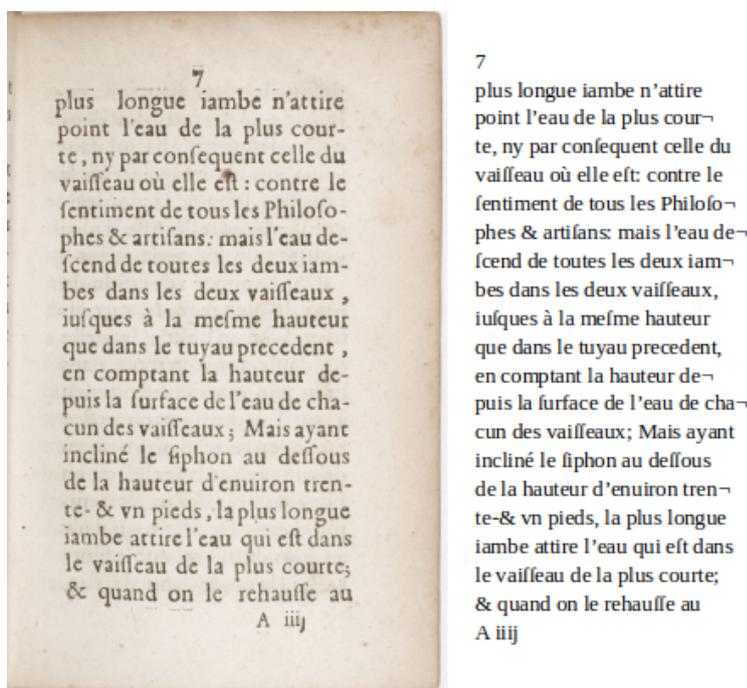


FIGURE 1 – Numérisation de la page 15 des *Experiences Nouvelles touchant le vide...* de Pascal (1647) présentée avec sa transcription diplomatique.

Rassemblé et transcrit par (Gabay, 2019), le corpus de travail est une sélection de certaines œuvres françaises du XVII<sup>e</sup> siècle décrites dans la table 1. Notre corpus environ 6 000 lignes, 30 000 mots et 15 0000 caractères. Un exemple de numérisation est proposé dans la figure 1.

i)				ii)			
Identifiant	Nb lignes	Nb mots	Nb caractères	Identifiant	Nb lignes	Nb mots	Nb caractères
Bossuet-1683	27	770	4 128	Papin-1682	23	548	2 230
Chapelain-1656	28	753	4 735	Pascal-1647	39	776	3 568
Ellain-1606	22	618	3 168	Sales-1641	25	618	3 915
Gournay-1622	31	825	4 284	Viau-1623	33	852	4 055

TABLE 2 – Description des sous-corpus dédiés à i) l’apprentissage des modèles de langue et ii) l’océrisation et l’évaluation de la qualité des sorties d’OCR.

Les variétés thématique et diachronique du corpus ainsi que les transcriptions diplomatiques de grande qualité permettent de le considérer non seulement comme un premier laboratoire privilégié pour l’étude de l’OCR mais aussi comme représentant de cet état de langue. Ainsi les œuvres de Bossuet, Chapelain, Ellain et Gournay et les œuvres de Papin, Pascal, Sales et Viau constituent-elles deux sous-corpus : les transcriptions des premières permettant l’apprentissage des modèles de langues et les images et les transcriptions des secondes l’application des logiciels d’OCR et la mesure du *CER* respectivement (voir la table 2).

## 3.2 Océrisation

Le corpus dédié à l’océrisation est composé de 120 pages numérisées, toutes avec une résolution de 400dpi. Afin de réaliser l’océrisation de ces images, deux logiciels ont été utilisés :

- Kraken, version 2.0.8<sup>5</sup> (voir (Kiessling, 2019));
- Tesseract, version 0.3.3<sup>6</sup> (voir (Smith, 2007)).

**Pré-traitement des images** Pour appliquer ses modèles de reconnaissance de caractères, Kraken prend en entrée des *lignes* binarisées (un pixel ne peut être que blanc ou noir) alors que Tesseract peut admettre des pages entières en nuances de gris ou même en couleurs. Dans un souci d’unité, et puisque Kraken est plus restrictif que Tesseract, toutes les images ont été segmentées et binarisées en utilisant les modules dédiés de Kraken. Ceci constitue un biais au regard des performances de Tesseract ; néanmoins, et selon notre hypothèse, le taux d’erreur devrait évoluer dans le même sens que les métriques d’estimation de l’étude.

**Application des modèles** L’objectif étant d’observer dans quelle mesure les modèles de langue peuvent être de bons indicateurs de la qualité d’une sortie d’OCR, l’utilisation de plusieurs modèles, adaptés ou non aux documents de l’étude, apparaît primordiale. Pour ce faire, les modèles de Kraken (anglais contemporain) et de Tesseract (français contemporain) ont été appliqués aux lignes segmentées et binarisées ainsi qu’un modèle Kraken appris sur ces mêmes données<sup>7</sup>. On dispose alors *a priori* de deux modèles non adaptés aux documents de l’étude<sup>8</sup> et d’un modèle suradapté à ces documents (puisque appris sur ceux-ci). L’hypothèse que nous faisons est que l’agrégation des proba-

5. <http://kraken.re/> et <https://pypi.org/project/kraken/>

6. <https://pypi.org/project/pytesseract/>

7. <https://github.com/e-ditiones/OCR17>

8. Par exemple, le <f> ne fait pas partie du vocabulaire des modèles de Kraken (anglais) et de Tesseract (français).

bilités offertes par les modèles de langue sur les sorties des modèles de Kraken et de Tesseract sera plus faible que sur les sorties du modèle Kraken appris sur ces mêmes données.

Lignes Kraken	CER	Lignes Tesseract	CER	Lignes Kraken 17	CER
	100,0 %		100,0 %	–	100,0 %
plus longue iambe n’attire	3,8 %	plus Jongue iambe n’attire	7,6 %	plus longue iambe n’attire	0 %
point lcau de la plus courte, ny par confequent celle du	7,4 %	point l’eau de la plus courte, ny par confequent celle du	3,7 %	point l’eau de la plus courte, ny par confequent celle du	0 %
vaiffeau oi elle ef : contre le	3,4 %	vaiffeau où elle et : contre le	3,4 %	vaiffeau ou elle elt : contre le	0 %
fentiment de tous les Philofo-	16,6 %	vaiffeau où elle et : contre le	10 %	vaiffeau ou elle elt : contre le	6,6 %
pbes artisans : maislcau de-	6,8 %	fentiment de tous les Philofo-	6,8 %	fentiment de tous les Philofo-	0 %
fcend de toutes lcs dcuxiam-	14,8 %	pbes & artisans : mais l’eau de-	7,4 %	pbes & artisans. mais l’eau de-	0 %
bes dans les dcux vaiffeaux,	14,2 %	fcend de toutes les deuxiam-	7,1 %	fcend de toutes les deux iam-	0 %
iufques a la mefme hauteur	11,1 %	bes dans les deux vaiffeaux ,	7,4 %	bes dans les deux vaiffeaux,	0 %
que dans le tuyau precedent,	11,5 %	iufques à la mefme hauteur	7,6 %	iufques à la mefme hauteur	3,8 %
en comptant la hauteur dec-	3,7 %	que dans le tuyau precedent ,	0 %	que dans le tuyau precedent,	0 %
puis la furface deleau de cha-	13,6 %	en comptant la hauteur de-	4,5 %	en comptant la hauteur de-	0 %
cun des vaiffeaux ; Mais ayant	12,9 %	puis la furface de l’eau de cha-	6,4 %	puis la furface de l’eau de cha-	0 %
inclin le fiphon au deffous	10,7 %	cun des vaiffeaux ; Maisayant	10,7 %	cun des vaiffeaux ; Mais ayant	0 %
dc la hauteur dcnuiron tren-	17,8 %	incliné le fiphon au deffous	10,7 %	incliné le fiphon au deffous	3,7 %
te- vn picds, lapluslongue	11,5 %	de la hauteur d’enuiron tren-	3,5 %	de la hauteur d’enuiron tren-	0 %
iambe attirclcau qui eft dans	11,5 %	te- & vn picds, la plus longue	3,8 %	te & vn pieds, la plus longue	0 %
le vaiffeau de la plus courte ;	16,1 %	jambe attire l’eau qui eft dans	9,6 %	iambe attire l’eau qui eft dans	0 %
quand on le rehauffe au	6,8 %	le vaiffeau de la plus courtc ;	10,3 %	e vaiffeau de la plus courte ;	3,4 %
A iii	8,6 %	& quand on le rehauffe au	8,6 %	& quand on le rehauffe au	0 %
	16,6 %	À ill	66,6 %	A iii	16,6 %

TABLE 3 – Sorties des trois modèles d’OCR pour la 15<sup>e</sup> page des *Experiences Nouvelles touchant le vide...* de Pascal (1647). De gauche à droite : modèle Kraken (anglais), modèle Tesseract (français) et modèle Kraken (français du XVII<sup>e</sup> siècle).

La table 3 présente un exemple d’utilisation des trois modèles d’OCR sélectionnés. Les lignes affichées sont les sorties des modèles et les CER ont été calculés face à la référence (présentée dans la figure 1).

### 3.3 Apprentissage des modèles de langue

Deux types de modèles de langue (au grain caractère) ont été appris sur le sous-corpus dédié (voir la table 2) qui compte 121 caractères différents :

- des modèles de langue à probabilités conditionnelles, appris comme la probabilité d’observer un caractère sachant une séquence de caractères (un historique) ;
- des modèles de langue appris par des réseaux de neurones (LSTM et biLSTM).

Le premier type de modèles constitue une *baseline* puisque ces modèles sont simplement construits en comptant, par fenêtre glissante sur le corpus d’apprentissage, le nombre d’occurrences du caractère suivant la séquence de caractères contenue dans la fenêtre glissante. Ces occurrences absolues sont ensuite divisées par la somme des occurrences de tous les caractères suivants cette séquence et sont utilisées comme des probabilités, puisque contenues dans l’intervalle [0; 1].

Les modèles de langue LSTM et biLSTM ont été choisis pour confronter aux modèles de la *baseline* des modèles appris par réseaux de neurones, en l’occurrence des réseaux de neurones récurrents. Ces modèles ont été appris en utilisant les modèles séquentiels de la librairie Python *keras*. Un *mapping* du vocabulaire est d’abord réalisé en prétraitement<sup>9</sup>. Les réseaux LSTM et biLSTM contiennent tous une couche *LSTM* et aux réseaux biLSTM est ajoutée une couche *Bidirectional* ; l’hypothèse étant ici

9. À chaque élément du vocabulaire (entendu comme l’ensemble des caractères différents) est associé un entier dans une table.

que i) tout caractère ne peut suivre tout autre caractère et que ii) tout caractère ne peut être précédé de tout autre caractère. Enfin, la fonction *softmax* est utilisée comme fonction d'activation. Ces modèles de langue ont été appris sur des séquences de  $n$  caractères, pour  $n$  variant de 2 à 10 et le nombre d'époques pour chaque apprentissage est 100. Finalement, on dispose donc de  $3 * 9 = 27$  modèles de langue pour tester l'estimation de la qualité des sorties des logiciels d'OCR. Le nombre de caractères dans le vocabulaire de ces modèles de langue est de 121.

## 3.4 Métriques d'estimation

### 3.4.1 Préambules

**Calcul du CER** Le CER est calculé entre une suite de caractères de référence et une suite à tester comme la somme des insertions, délétions et substitutions divisée par le nombre total de caractères de la chaîne de référence. Il peut être supérieur à 1 si le nombre d'insertions est particulièrement élevé.

**Probabilités des modèles de langue** Les modèles de langue permettent de disposer de la probabilité qu'un caractère donné suive une certaine séquence de caractères. Si une sortie de logiciel d'OCR est parcourue par une fenêtre glissante à partir de laquelle est renvoyée une séquence de caractères et le caractère suivant cette séquence, pour une sortie d'OCR on dispose d'une suite d'au plus  $C - n$  probabilités, avec  $C$  le nombre total de caractères et  $n$  la taille de la fenêtre glissante en caractères. *Au plus* car il est possible que certains caractères fournis par le modèle d'OCR n'aient pas été rencontrés dans le corpus d'apprentissage du modèle de langue <sup>10</sup>.

On cherche donc à agréger ces probabilités, pour chaque document du corpus ocrisé, dans l'objectif que ces agrégats soient corrélés au CER qu'on peut calculer grâce aux transcriptions. Il s'agit de calculer d'autres métriques ne nécessitant pas de vérité de terrain (à partir des probabilités fournies par les modèles de langue) et de valider ou réfuter la pertinence de leur estimation de la qualité d'une sortie d'OCR face à une métrique de référence, le CER.

### 3.4.2 Agrégations des probabilités des modèles de langue

**La somme des probabilités** Une première métrique peut être la somme des probabilités renvoyées par les modèles de langue. Sous réserve que les modèles de langue sont bien des distributions de probabilités, la somme des probabilités d'une suite de caractères correspondant à du texte est de 1 alors qu'elle ne peut l'être dans le cas contraire. Ainsi, pour une sortie d'OCR, on a :

$$S = \sum_{i=n+1}^{C-n} P_{LM}(c_i | h_{n,i})$$

Avec  $P_{LM}$  la probabilité renvoyée par un modèle de langue  $LM$ ,  $n$  la taille de la fenêtre glissante en caractères,  $C$  le nombre total de caractères de la sortie d'OCR,  $c_i$  le  $i^e$  caractère de la sortie d'OCR et  $h_{n,i}$  l'historique de  $n$  caractères du caractère  $c_i$ .

10. Par exemple, Kraken (anglais) et Tesseract (français), appris sur des documents contemporains, ont dans leur vocabulaire le symbole € et peuvent le proposer dans leur ocrisation. Pour un modèle de langue appris sur des données textuelles françaises du XVII<sup>e</sup> siècle, ce symbole n'existe pas.

**Le produit des probabilités** Le produit des probabilités peut aussi constituer une autre métrique d'estimation. Il rend compte de la probabilité d'une suite de caractères selon l'hypothèse, ici réductrice, de l'indépendance. Il est défini comme :

$$Pr = \prod_{i=n+1}^{C-n} P_{LM}(c_i|h_{n,i})$$

**La perplexité** Plus couramment utilisée pour juger de la qualité d'un modèle de langue, la perplexité est la probabilité inverse de la sortie d'OCR normalisée par son nombre de caractères. Puisqu'elle mesure la distance entre la fonction de probabilité et les données de l'ensemble de test, elle semble pertinente à tester comme métrique d'estimation. Elle est définie comme :

$$PP = \frac{1}{\left(\prod_{i=n+1}^{C-n} P_{LM}(c_i|h_{n,i})\right)^{\frac{1}{C-n}}}$$

**La log-perplexité** Enfin, la log-perplexité peut aussi constituer une métrique d'estimation ; (Chen *et al.*, 1998) montrent qu'elle aussi est corrélée au *WER* dans le domaine de la reconnaissance de la parole.

### 3.4.3 Échelles de calcul des agrégations

Les agrégations des probabilités précitées peuvent être calculées à plusieurs échelles : celle de l'œuvre, de la page, de la ligne ou encore du mot. Puisque la perplexité *PP* est calculée comme l'inverse d'une racine *n*-ième, elle tend vers 1 à mesure que le nombre total de caractères  $C - n$  grandit. Plus le nombre de caractères sur lesquels elle est calculée grandit, moins elle est informative. Les estimateurs de qualité d'océrisation des pages sont donc calculés comme la moyenne des agrégations des probabilités calculées à l'échelle du mot<sup>11</sup>.

Notons que réaliser une *moyenne* constitue un biais important. Les données textuelles issues d'OCR n'ont pas une qualité homogène pour une même œuvre ou une même page ; la moyenne efface ces disparités pourtant essentielles à soulever. D'autre part, certains mots comportent trop peu de caractères pour que le modèle de langue puisse leur calculer une probabilité. Certains passages sont ignorés et la moyenne ne le traduit pas.

## 4 Expérimentations et résultats

Pour calculer les métriques d'estimation de la qualité de l'OCR sur une page du corpus, on calcule ces métriques pour chaque mot de la page et on en fait la moyenne. Afin de confirmer ou réfuter l'intérêt de ces métriques, un calcul de corrélation Pearson<sup>12</sup> est réalisé avec le *CER*. Si une ou

11. La tokenisation est réalisée par une simple segmentation par l'espace des chaînes de caractères.

12. En tant que normalisation des covariances, le coefficient de corrélation exprime à quel point deux variables sont liées. Ce coefficient étant une normalisation, il appartient à l'intervalle  $[-1; 1]$  ; les corrélations positives indiquent que les deux variables évoluent dans le même sens et les corrélations négatives qu'elles évoluent dans un sens opposé. Plus une corrélation est proche de 1 ou  $-1$ , plus le lien entre les deux variables est fort ; au contraire, plus la corrélation est proche de 0, plus ce lien se dissipe.

plusieurs métriques est corrélée-s significativement au *CER*, on peut conclure que l'apprentissage d'un modèle de langue sur des données du français du XVII<sup>e</sup> siècle et l'utilisation de ses probabilités pour estimer la qualité d'une sortie d'OCR sur des documents de la même période sont justifiés.

## 4.1 Préambule à l'analyse des corrélations

Métriques	Variations	Signes des corrélations avec le <i>CER</i>
<i>CER</i>	↗	
<i>S</i>	↘	-
<i>Pr</i>	↘	-
$Pr^{\frac{1}{C-n}}$	↘	-
$PP = \frac{1}{Pr^{\frac{1}{C-n}}}$	↗	+
$\log(PP)$	↗	+

TABLE 4 – Variations et signes des corrélations avec le *CER* des métriques d'estimation pour un nombre d'erreurs qui augmente.

La table 4 expose la variation des métriques d'estimation et le signe de leur corrélation avec le *CER* pour un nombre d'erreurs d'océrisation qui augmente. L'hypothèse est que les modèles de langue fournissent des probabilités (voir le paragraphe 3.4.1) plus élevées face à une sortie d'OCR sans erreur (du *texte*) et des probabilités plus faibles face à une sortie d'OCR avec erreurs (du *non-texte*). Ainsi, pour valider les métriques comme estimateurs pertinents, les corrélations entre le *CER* et la somme et le produit des probabilités doivent être négatives alors qu'elles doivent être positives entre le *CER* et la perplexité et la log-perplexité.

## 4.2 Corrélations entre le *CER* et les métriques d'estimation

Le sous-corpus dédié à l'OCR est composé de 118 pages. On peut donc, pour les sorties des trois modèles d'OCR utilisés et pour les trois types de modèles de langue, calculer les métriques d'estimation et le *CER*, et ce pour chaque page du corpus.

OCR : Kraken (français contemporain). ML : probabilités conditionnelles.									
	S		Pr		PP		log(PP)		
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value	
n=2	-0,063	0,496	0,111	0,226	-0,016	0,862	-0,013	0,884	
n=3	-0,098	0,287	0,073	0,426	0,005	0,955	0,021	0,820	
n=4	-0,073	0,428	-0,043	0,642	-0,010	0,913	-0,014	0,879	
n=5	-0,137	0,135	-0,043	0,638	-0,015	0,868	-0,026	0,780	
n=6	-0,093	0,314	0,000	0,996	0,067	0,466	0,059	0,522	
n=7	-0,035	0,708	-0,032	0,728	0,130	0,157	0,117	0,205	
n=8	-0,064	0,485	-0,074	0,420	0,043	0,643	0,054	0,560	
n=9	-0,057	0,538	-0,012	0,898	0,018	0,846	0,021	0,821	
n=10	-0,046	0,615	-0,023	0,806	0,024	0,794	0,026	0,780	
OCR : Tesseract (anglais contemporain). ML : probabilités conditionnelles.									
	S		Pr		PP		log(PP)		
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value	
n=2	-0,004	0,968	0,158	<b>0,086</b>	0,113	0,221	0,006	0,952	

n=3	-0,130	0,156	0,009	0,920	-0,003	0,976	0,056	0,540
n=4	-0,124	0,178	-0,005	0,960	0,016	0,866	0,060	0,518
n=5	-0,158	<b>0,084</b>	-0,070	0,449	0,134	0,143	0,158	<b>0,085</b>
n=6	-0,138	0,133	-0,054	0,556	0,180	<b>0,049</b>	0,188	<b>0,040</b>
n=7	-0,100	0,278	-0,027	0,773	0,093	0,313	0,084	0,359
n=8	-0,055	0,547	-0,008	0,930	-0,006	0,949	-0,008	0,928
n=9	-0,054	0,554	-0,083	0,366	0,096	0,299	0,095	0,300
n=10	-0,024	0,796	-0,212	<b>0,020</b>	0,228	<b>0,012</b>	0,187	<b>0,041</b>

OCR : Kraken (français XVII<sup>e</sup>). ML : probabilités conditionnelles.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,052	0,572	0,129	0,162	0,238	<b>0,009</b>	0,040	0,663
n=3	-0,080	0,384	0,087	0,343	-0,003	0,970	0,030	0,742
n=4	-0,041	0,654	-0,010	0,916	0,012	0,894	0,031	0,739
n=5	-0,111	0,225	-0,072	0,437	0,039	0,670	0,069	0,452
n=6	-0,135	0,142	-0,055	0,551	0,138	0,133	0,143	0,120
n=7	-0,086	0,348	-0,030	0,746	0,063	0,496	0,075	0,414
n=8	-0,096	0,296	-0,030	0,743	-0,045	0,622	-0,045	0,625
n=9	-0,052	0,574	-0,017	0,857	-0,015	0,874	-0,029	0,757
n=10	-0,021	0,817	-0,039	0,669	0,121	0,189	0,097	0,291

OCR : Kraken (français contemporain). ML : LSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,059	0,520	0,094	0,305	0,046	0,615	0,000	0,999
n=3	-0,078	0,395	0,078	0,397	-0,031	0,739	-0,011	0,902
n=4	-0,094	0,306	-0,070	0,446	0,113	0,221	-0,111	0,227
n=5	-0,108	0,240	-0,048	0,600	-0,061	0,511	-0,077	0,404
n=6	-0,039	0,675	0,184	0,045	0,055	0,552	-0,092	0,320
n=7	0,198	<b>0,030</b>	-0,049	0,595	-0,051	0,578	0,035	0,702
n=8	-0,034	0,709	0,026	0,781	-0,017	0,854	0,058	0,530
n=9	-0,055	0,549	-0,016	0,860	-0,062	0,499	0,076	0,407
n=10	0,063	0,492	-0,036	0,695	-0,055	0,550	-0,061	0,506

OCR : Tesseract (anglais contemporain). ML : LSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,010	0,911	0,138	0,131	-0,032	0,730	-0,081	0,377
n=3	-0,132	0,151	-0,046	0,621	-0,004	0,962	0,133	0,148
n=4	-0,085	0,354	-0,053	0,567	0,003	0,972	-0,009	0,926
n=5	-0,096	0,298	-0,080	0,383	-0,022	0,808	0,053	0,568
n=6	-0,107	0,245	0,034	0,716	-0,010	0,915	-0,038	0,683
n=7	-0,116	0,208	-0,079	0,390	0,106	0,250	0,007	0,938
n=8	0,043	0,640	-0,026	0,776	-0,034	0,711	-0,025	0,788
n=9	-0,044	0,636	0,024	0,791	-0,032	0,732	0,029	0,756
n=10	-0,025	0,788	-0,072	0,432	0,012	0,895	0,073	0,426

OCR : Kraken (français XVII<sup>e</sup>). ML : LSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,055	0,552	0,168	<b>0,066</b>	-0,006	0,944	-0,049	0,596
n=3	-0,065	0,482	0,114	0,215	-0,023	0,804	-0,062	0,503
n=4	-0,058	0,526	-0,030	0,741	-0,026	0,777	-0,024	0,793
n=5	-0,069	0,455	-0,052	0,576	-0,012	0,898	-0,005	0,953
n=6	-0,083	0,366	0,045	0,623	-0,008	0,930	0,017	0,856
n=7	-0,067	0,465	-0,049	0,595	-0,027	0,773	-0,056	0,541
n=8	-0,104	0,258	-0,041	0,653	-0,030	0,745	-0,029	0,751
n=9	-0,023	0,805	-0,036	0,697	-0,022	0,809	-0,034	0,710
n=10	0,181	<b>0,048</b>	-0,059	0,520	-0,006	0,947	-0,012	0,898

OCR : Kraken (français contemporain). ML : biLSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,042	0,645	0,115	0,213	-0,040	0,661	-0,043	0,641
n=3	-0,098	0,289	0,160	<b>0,081</b>	-0,096	0,295	-0,077	0,404
n=4	-0,091	0,320	-0,087	0,346	0,085	0,357	-0,126	0,170

n=5	-0,019	0,837	-0,085	0,354	-0,049	0,595	-0,013	0,891
n=6	-0,076	0,411	0,131	0,154	-0,040	0,662	-0,087	0,345
n=7	0,010	0,914	-0,105	0,255	-0,058	0,529	0,009	0,925
n=8	-0,053	0,564	-0,085	0,357	0,623	<b>0,001</b>	0,053	0,563
n=9	-0,070	0,446	-0,060	0,517	-0,024	0,794	-0,054	0,560
n=10	0,084	0,361	-0,028	0,758	-0,033	0,722	-0,059	0,521

OCR : Tesseract (anglais contemporain). ML : biLSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	0,013	0,886	0,184	<b>0,044</b>	0,000	1,000	-0,117	0,203
n=3	-0,137	0,137	-0,022	0,816	0,117	0,203	0,157	<b>0,087</b>
n=4	-0,040	0,663	0,011	0,909	-0,025	0,788	-0,051	0,577
n=5	-0,079	0,389	-0,055	0,547	-0,006	0,946	0,023	0,800
n=6	-0,058	0,530	-0,012	0,893	-0,045	0,627	-0,098	0,287
n=7	-0,064	0,485	-0,013	0,885	-0,007	0,943	-0,093	0,314
n=8	0,036	0,696	-0,023	0,801	0,051	0,580	-0,036	0,694
n=9	-0,053	0,566	-0,033	0,724	0,007	0,936	0,050	0,586
n=10	-0,029	0,754	0,029	0,756	-0,024	0,797	0,027	0,773

OCR : Kraken (français XVII<sup>e</sup>). ML : biLSTM.

	S		Pr		PP		log(PP)	
	corrélation	p-value	corrélation	p-value	corrélation	p-value	corrélation	p-value
n=2	-0,061	0,508	0,146	0,113	0,002	0,985	-0,060	0,513
n=3	-0,075	0,413	0,101	0,273	-0,026	0,779	-0,111	0,228
n=4	-0,048	0,601	-0,045	0,622	-0,011	0,909	-0,078	0,396
n=5	-0,040	0,661	-0,106	0,250	-0,038	0,681	-0,014	0,878
n=6	-0,045	0,622	-0,010	0,917	-0,048	0,605	-0,041	0,660
n=7	-0,106	0,251	0,064	0,489	-0,005	0,955	-0,021	0,823
n=8	-0,086	0,350	-0,057	0,539	0,022	0,815	-0,076	0,410
n=9	-0,017	0,855	-0,033	0,721	-0,007	0,940	0,011	0,908
n=10	0,202	<b>0,027</b>	-0,025	0,786	-0,027	0,767	0,012	0,898

TABLE 5 – Corrélations et *p-values* calculées entre les métriques d'estimation et le *CER*.

La table 5 montre les corrélations et les *p-values* calculées entre les métriques d'estimation et les *CER*. Une corrélation n'est toutefois significative que si la *p-value* est inférieure à un seuil, ceci traduisant que la relation de corrélation a peu de chances d'être due au hasard; il s'agit d'un test de corrélation. Nous choisissons ici le seuil de 0, 1 qui rend compte d'une faible présomption contre l'hypothèse nulle. Les résultats de la table 5 montre des *p-values* presque toutes supérieures à ce seuil, ce qui signifie que s'il y a corrélation, elle n'est pas significative. Ces résultats semblent donc réfuter l'hypothèse initiale selon laquelle les probabilités des modèles de langue auraient pu être agrégées pour se substituer à un *CER* exigeant une vérité de terrain.

### 4.3 Les modèles de langue sont-ils inadaptés ?

	ML probabilités conditionnelles	ML LSTM	ML biLSTM
n=2	90	14721	257646757092
n=3	126	1010690	235913940342
n=4	426	318251055	221055920422
n=5	1091	723946838	211044617070
n=6	1978	690749546	204520506752
n=7	2801	669397958	200184237186
n=8	3510	655634987	1161841181775
n=9	3940	647905538	13807745026062
n=10	4205	643364471	14481238375005

TABLE 6 – Moyennes des perplexités des modèles de langue sur le sous-corpus de test.

Les résultats précédents suggèrent que i) soit les modèles de langue sont de mauvaise qualité, ii)

soit le corpus de l'étude présente des spécificités particulières ou iii) soit les deux raisons précitées concourent à cette impasse.

Les modèles de langue ont été appris sur les œuvres de Bossuet, Chapelin, Ellain et Gournay. On peut donc les évaluer en calculant leur perplexité sur le sous-corpus des œuvres de Papin, Pascal, Sales et Viau. La table 6 présente les moyennes des perplexités des modèles de langue de l'étude calculées sur les vérités de terrain. Les modèles de langue LSTM et biLSTM présentent des perplexités aberrantes (ils sont non adaptés à la tâche) alors que seuls les modèles de langue à probabilités conditionnelles, pour  $n \in [2; 4]$ , présentent une meilleure qualité. Nous concluons donc que la mauvaise qualité des modèles de langue explique la non corrélation entre les estimateurs et le *CER*.

## 5 Conclusion

La mauvaise qualité des modèles de langue ne permet pas de valider ou réfuter notre hypothèse, selon laquelle agréger les probabilités des modèles de langue permettrait d'estimer la qualité d'une sortie d'OCR. Pour en faire l'expérience, il s'agirait de renouveler ces tests avec un ensemble plus vaste de transcriptions d'imprimés du XVII<sup>e</sup> siècle. Nous cherchions à proposer une alternative au manque de vérités de terrain mais nous constatons qu'un ensemble de 108 pages (16 315 mots) est insuffisant. Si cela ne contredit pas l'éventuelle pertinence des estimateurs envisagés, un ensemble conséquent de données textuelles en français du XVII<sup>e</sup> siècle reste nécessaire au bon apprentissage des modèles langue. Nous rassemblerons donc plus de données textuelles en français du XVII<sup>e</sup> siècle pour reconduire l'expérience avec des modèles de langue de meilleure qualité.

Les programmes, en Python 3, sont mis à disposition sur : <https://github.com/jbtanguy/RECITAL2020>.

## Remerciements

Ce travail n'aurait pas été possible sans l'aide de Simon Gabay, Gaël Lejeune et Alice Millour.

## Références

- BERG-KIRKPATRICK T. & KLEIN D. (2014). Improved Typesetting Models for Historical OCR. In K. TOUTANOVA & H. WU, Édts., *Actes de 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, volume 2, p. 118–123, Baltimore, Maryland, États-Unis : Association for Computational Linguistics. Anthologie ACL : [P14-2020](#).
- BERMES E. (2020). *Le numérique en bibliothèque : naissance d'un patrimoine : l'exemple de la Bibliothèque nationale de France (1997-2019)*. Thèse de doctorat, Paris, Ecole nationale des chartes.
- BREUEL T. M., UL-HASAN A., AL-AZAWI M. A. & SHAFAIT F. (2013). High-performance OCR for printed English and Fraktur using LSTM networks. In *Actes de 12th International Conference on Document Analysis and Recognition, ICDAR'13*, p. 683–687, Washington, DC, États-Unis : IEEE IEEE Computer Society.

- CHEN S. F., BEEFERMAN D. & ROSENFELD R. (1998). Evaluation metrics for language models. In *Actes de DARPA Broadcast News Transcription and Understanding Workshop*, p. 275–280, Lansdowne, Virginia, États-Unis : Carnegie Mellon University.
- GABAY S. (2019). OCRising 17th French prints. <https://editiones.hypotheses.org/1958>.
- GUPTA A., GUTIERREZ-OSUNA R., CHRISTY M., CAPITANU B., AUVIL L., GRUMBACH L., FURUTA R. & MANDELL L. (2015). Automatic assessment of OCR quality in historical documents. In *Actes de Twenty-Ninth AAAI Conference on Artificial Intelligence*, p. 1735–1741, Austin, Texas, États-Unis.
- KIESSLING B. (2019). Kraken—an universal text recognizer for the humanities. In ADHO, Éd., *Actes de Digital Humanities Conference 2019 - DH2019*, Utrecht, Pays-Bas.
- LEJEUNE G. & ABIVEN K. (2019). Analyse automatique de documents anciens : tirer parti d'un corpus incomplet, hétérogène et bruité. *Information Retrieval, Document and Semantic Web*, **19**(1).
- SMITH R. (2007). An overview of the Tesseract OCR engine. In *Actes de Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, p. 629–633, Parana, Brésil : IEEE.
- SPRINGMANN U., FINK F. & SCHULZ K. U. (2016). Automatic quality evaluation and (semi-) automatic improvement of OCR models for historical printings. arXiv preprint : [1606.05157](https://arxiv.org/abs/1606.05157).
- SPRINGMANN U. & LÜDELING A. (2016). OCR of historical printings with an application to building diachronic corpora : A case study using the RIDGES herbal corpus. arXiv preprint : [1608.02153](https://arxiv.org/abs/1608.02153).
- SPRINGMANN U., NAJOCK D., MORGENROTH H., SCHMID H., GOTSCHAREK A. & FINK F. (2014). OCR of historical printings of Latin texts : problems, prospects, progress. In *Actes de First International Conference on Digital Access to Textual Cultural Heritage (DATeCH'14)*, p. 71–75, New York, NY, États-Unis : Association for Computing Machinery.
- SPRINGMANN U., REUL C., DIPPER S. & BAITER J. (2018). Ground Truth for training OCR engines on historical documents in German Fraktur and Early Modern Latin. arXiv preprint : [1809.05501](https://arxiv.org/abs/1809.05501).
- UL-HASAN A., BUKHARI S. S. & DENGEL A. (2016). Ocroact : A sequence learning ocr system trained on isolated characters. In *Actes de 12th IAPR Workshop on Document Analysis Systems (DAS)*, p. 174–179, Santorini, Grèce : IEEE.
- VAMVAKAS G., GATOS B., STAMATOPOULOS N. & PERANTONIS S. J. (2008). A complete optical character recognition methodology for historical documents. In *Actes de Eighth IAPR International Workshop on Document Analysis Systems*, p. 525–532, Nara, Japon : IEEE.
- WICK C., REUL C. & PUPPE F. (2018). Comparison of OCR Accuracy on Early Printed Books using the Open Source Engines Calamari and OCRopus. *ACM/IEEE Joint Conference on Digital Libraries 2018 (JCDL 2018)*, **33**(1), 79–96.