Exploiter des modèles de langue pour évaluer des sorties de logiciels d'OCR pour des documents français du XVII^{e} siècle

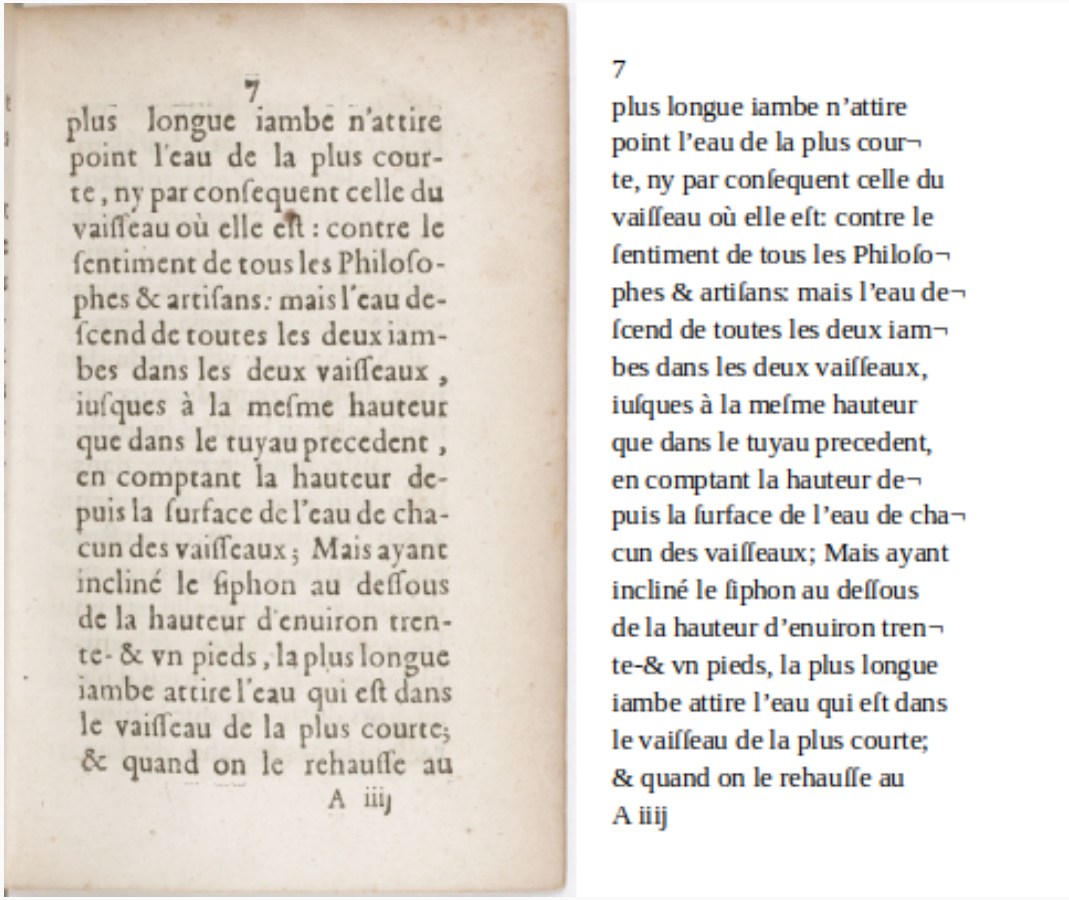
Jean-Baptiste Tanguy

16 avril 2020

LabEx OBVIL, Sorbonne Université

Plan de la présentation

Introduction



7

plus longue iambe n'attire point l'eau de la plus courte, ny par conſequent celle du vaiſſeau où elle eſt : contre le ſentiment de tous les Philoſophes & artiſans: mais l'eau deſcend de toutes les deux iambes dans les deux vaiſſeaux, iuſques à la meſme hauteur que dans le tuyau precedent, en comptant la hauteur depuis la ſurface de l'eau de chacun des vaiſſeaux; Mais ayant incliné le ſiphon au deſſous de la hauteur d'enuiron trente-& vn pieds, la plus longue iambe attire l'eau qui eſt dans le vaiſſeau de la plus courte; & quand on le rehauſſe au

A iiij

7
plus longue iambe n'attire
point l'eau de la plus cour¬
te, ny par conſequent celle du
vaiſſeau où elle eſt: contre le
ſentiment de tous les Philoſo¬
phes & artiſans: mais l'eau de¬
ſcend de toutes les deux iam¬
bes dans les deux vaiſſeaux,
iuſques à la meſme hauteur
que dans le tuyau precedent,
en comptant la hauteur de¬
puis la ſurface de l'eau de cha¬
cun des vaiſſeaux; Mais ayant
incliné le ſiphon au deſſous
de la hauteur d'enuiron tren¬
te-& vn pieds, la plus longue
iambe attire l'eau qui eſt dans
le vaiſſeau de la plus courte;
& quand on le rehauſſe au
A iiij

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

Introduction

Lignes Kraken	CER	Lignes Tesseract	CER
plus longue iambe n'attire	3,8 %	plus Jongue iambe n'attire	7,6 %
point lcau de la plus cour-	7,4 %	point l'eau de la plus cour-	3,7 %
te, ny par confequent celle du	3,4 %	te, ny par confequent celle du	3,4 %
vaifeau oi elle ef : contre le	16,6 %	vaiffeau où elle et : contre le	10 %
fentiment de tous les Philofo-	6,8 %	fentiment de tous les Philofo-	6,8 %
phes artifans : maislcau de-	14,8 %	phes & artifans : mais l'eau de-	7,4 %
fcend de toutes lcs dcuxiam-	14,2 %	fcend de toutes les deuxiam-	7,1 %
bes dans les dcux vaiffeaux,	11,1 %	bes dans les deux vaiffeaux ,	7,4 %
iufques a la mefme hauteur	11,5%	iufques à la mefme hauteur	7,6 %
que dans le tuyau preccdent,	3,7 %	que dans le tuyau precedent ,	0 %
en comptant la hautcur dec-	13,6 %	en comptant la hauteur de-	4,5%

Table 1 – Sorties d'OCR et *CER* de Kraken et Tesseract pour le début de la page 15 des *Experiences Nouvelles touchant le vide...* de Pascal (1647)

Introduction

Évaluation d'une sortie d'OCR : transcription diplomatique puis calcul du *CER*, avec :

$$CER = \frac{s + d + i}{C}$$

Transcription diplimatique (au moins pour état de langue français du XVII^{e}) :

- nécessite de l'expertise (philologie computationnelle) ;
- prend beaucoup de temps ;
- nécessaire à toute évaluation.

Comment évaluer la qualité d'une sortie d'OCR **sans** vérité de terrain ?

Évaluation non supervisée de sorties d'OCR

Ce qui se fait

Plusieurs voies :

- Exploiter des ressources lexicales (la *lexicalité* d'une sortie d'OCR) [Springmann *et al.*, 2016] ;
- Exploiter les valeurs de confiance des logiciels d'OCR [Springmann *et al.*, 2016] ;
- Exploiter les *bounding boxes* [Gupta *et al.*, 2015] ;
- (Reconnaissance de la parole) Exploiter les modèles de langue [Chen *et al.*, 1998].

Ce que nous faisons

Démarche :

- Apprentissage de modèles de langue (grain caractère) sur des données textuelles françaises du XVII^{e} siècle
- Application des modèles d'OCR sur un corpus de documents numérisés du XVII^{e} siècle
- Calcul des *CER* de ces sorties d'OCR (calculés avec les vérités de terrain)
- Calcul d'estimateurs de qualité utilisant les modèles de langue

Objectifs :

- Définir et calculer les estimatateurs de qualité d'OCR (exploitation des modèles de langue)
- Étudier leurs corrélations avec les *CER* (et les p-values)

Définition des estimateurs de qualité d'OCR

Comment utiliser les probabilités des modèles de langue ?

Les modèles de langue apprennent les probabilités que des caractères donnés suivent certaines séquences de caractères.

On peut donc :

- parcourir un texte par fenêtre glissante...
- ... récupérer la séquence de caractères contenue dans cette fenêtre ainsi que le caractère suivant...
- ... et récupérer la probabilité renvoyée par un modèle de langue pour que ce caractère suive cette séquence de caractères.

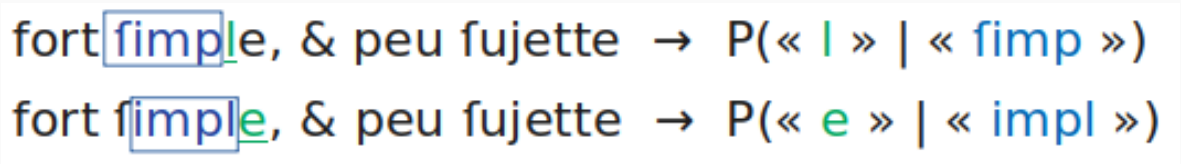


Figure 2 – Parcours d'un texte par fenêtre (n=4) glissante pour récupérer la probabilité d'un caractère sachant un histoire.

Comment utiliser les probabilités des modèles de langue ?

Hypothèse : ces probabilités peuvent être de bons indices pour estimer la qualité d'une sortie d'OCR

- Océrisation douteuse ⇒ suite de caractères qui n'est pas du texte ⇒ faibles probabilités
- Océrisation de qualité ⇒ suite de caractères formant du texte ⇒ fortes probabilités

Comment agréger les probabilités ?

La somme S, le produit Pr, la perplexité Pp et le log-perplexité $log(PP)$ sont calculés pour chaque ligne puis moyennés sur la page

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

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

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

$$log(PP)$$

Avec P_{LM} la probabilité renvoyée par un modèle de langue LM, n la taille de la fenêtre glissante, 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 des n caractères

Expériences et résultats

Les corpus

[Gabay, 2019] a rassemblé et transcrit plusieurs œuvres françaises du XVII^{e} siècle

i) Corpus pour l'apprentissage des modèles de langue

Identifiant	Nb lignes	Nb mots	Nb caractères
Bossuet-1683	27	770	4 128
Chapelain-1656	28	753	4 735
Ellain-1606	22	618	3 168
Gournay-1622	31	825	4 284

ii) Corpus pour l'océrisation et son évaluation

Identifiant	Nb lignes	Nb mots	Nb caractères
Papin-1682	23	548	2 230
Pascal-1647	39	776	3 568
Sales-1641	25	618	3 915
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 technologies utilisées

Pour l'OCR :

- [Kiessling, 2019] : Kraken, modèle pour l'anglais contemporain et modèle pour le français du XVII^{e} siècle
- [Smith, 2007] : Tesseract, modèle pour l'anglais contemporain

Pour les modèles de langue :

- modèles de langue à probabilités conditionnelles (fait maison)
- modèles de langue appris par des réseaux de neurones (LSTM et biLSTM)

Résultats (je vous épargne les tableaux)

Aucune corrélation entre les estimateurs et les *CER* ($pvalues \gg 0.05$), et ce pour :

- les deux logiciels d'OCR ;
- les trois modèles d'OCR ;
- les quatre estimateurs ;
- tous les modèles de langue (proba conditionnelles, LISTM, biLSTM… pour $n \in [\![2\ ;\ 10]\!]$.

Pourquoi ?

La perplexité peut être calculée pour évaluer les modèles de langue (sur une référence). Sur l'ensemble des transcriptions dont on dispose :

	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	1380774502606 2
n=10	4205	643364471	14481238375005

TABLE 3 – Moyennes des perplexités des modèles de langue sur le sous-corpus de test.

Pourquoi ?

- Une perplexité faible suggère que le modèle de langue est de qualité
- Le tableau précédent montre des valeurs **aberrantes**
- Sauf pour les modèles de langue à probabilités conditionnelles (fait maison) pour $n \in [\![2\ ;\ 4]\!]$

Conclusion

- La majorité des modèles de langue sont inadaptés (manque de données d'apprentissage ? construction naïve des réseaux ?)
- On dispose de beaucoup plus de données en français du XVII^{e} siècle pour reconduire le travail
- Objectif : isoler le problème (modèles de langue ? estimateurs ? corpus ?)

Annexe : construction actuelle des réseaux LSTM et biLSTM

```
X, y = encoded_sequences[:,:-1], encoded_sequences[:,-1]
sequences = [to_categorical(x, num_classes=vocab_size) for x in X] # one-hot representation
X = array(sequences)
y = to_categorical(y, num_classes=vocab_size) # one-hot representation
# d. Define the model
model = Sequential()
if self._bilstm == True:
    model.add(Bidirectional(LSTM(vocab_size, input_shape=(X.shape[1], X.shape[2]), return_sequences=True)))
    model.add(Bidirectional(LSTM(vocab_size)))
else:
    model.add(LSTM(vocab_size, input_shape=(X.shape[1], X.shape[2])))
model.add(Dense(vocab_size, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, y, epochs=100, verbose=2)
d = {'model': model, 'mapping': mapping}
return d
```

FIGURE 3 – Capture d'écran du code construisant les réseaux LSTM et biLSTM.


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.

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 e-prints.