

LE PROJET AGODA

Océrisation des débats parlementaires français de la Troisième République : problèmes, défis et perspectives

Aurélien Pellet et Marie Puren (MNSHS-Epitech)

25 avril 2022

Séminaire EPITECH/OMNSH

Contexte et objectifs

- Le projet AGODA

- Objectifs

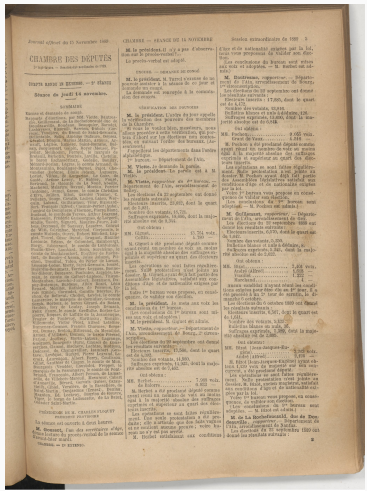
- En mode « preuve de concept »

Occurrer des documents imprimés

- Les problèmes que nous rencontrons

- Les solutions choisies

LES DÉBATS PARLEMENTAIRES DURANT LA TROISIÈME RÉPUBLIQUE



- Débats à la Chambre des députés (chambre basse du parlement) transcrits en détail dans le **Journal officiel de la République française. Débats parlementaires** (1881-1940)
- Disponible en ligne via **Gallica** (bibliothèque numérique de la Bibliothèque nationale de France)
- Difficile de travailler sur ce corpus, pourtant intéressant pour diverses disciplines (histoire, sociologie, science politique, linguistique)

Figure – Séance parlementaire du 14 novembre 1889

CONTEXTE ET OBJECTIFS

- AGODA : **A**nalyse sémantique et **G**raphes relationnels pour l'**O**uverture et l'étude des **D**ébats à l'**A**ssemblée nationale
- Projet financé par la Bibliothèque nationale de France pour une durée d'un an
- L'un des 5 projets-pilote soutenu par le **DataLab**

- Donner plus facilement accès aux retranscriptions anciennes des débats parlementaires
- Faciliter la recherche dans ce corpus
- Offrir de nouveaux modes de visualisation des documents

- Créer une plateforme de consultation
- Produire des données textuelles structurées et sémantiquement enrichies à partir de ces débats numérisés
- Contribuer à la conception d'un workflow adapté à l'analyse de gros corpus de documents historiques

Traitement d'une sous-partie du corpus : législature 1889-1893 soit **10418 images à traiter**

- Renouvellement partiel du personnel politique (boulangisme et scandale de Panama)
- Premières manifestations du Ralliement des catholiques à la République
- Tournant de la politique douanière (lois Méline)
- Essor du socialisme et du syndicalisme (Fourmies)
- Premiers attentats anarchistes

1. Extraire le texte des images (océration)
2. Annoter et enrichir sémantiquement le corpus
3. Publier un corpus éditorialisé

OCÉRISER DES DOCUMENTS IMPRIMÉS

Numérisation en masse de textes : comment avoir accès à/ traiter/ analyser leur contenu ?

- OCR : Optical Character Recognition ou Reconnaissance optique des caractères
- Traitement d'une image (texte numérisé) par un logiciel de reconnaissance de caractères
- Utilisation de l'IA : « traduction » de l'image en texte
 - Pages numérisées, puis transformées en lettres et en mots « discrets » (dénombrables : discontinus, séparés, distincts)
 - Possible de parcourir ces pages et de les « différencier » par période, genre, langue, année de publication, etc.

Tâche cruciale pour exploitation des documents historiques - et plus généralement pour traitement en masse des documents (Taille marché OCR estimée aux US **13,38 milliards de dollars US en 2025**)

- Récupération des textes océrésés via API Document de Gallica => mauvaise qualité de l'OCR
- Erreurs dues à la courbure de la page au niveau de la reliure + tâches, ombres...etc.

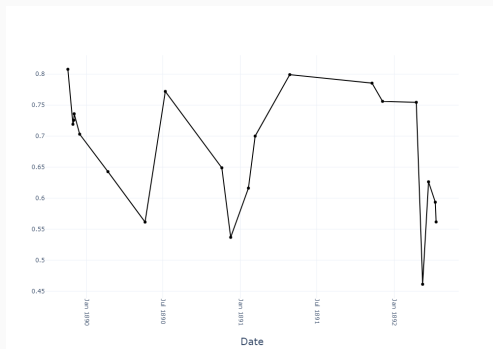
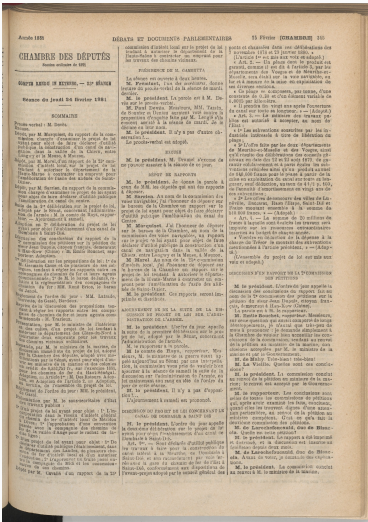


Figure – Evaluation de la qualité de l'OCR fourni par Gallica

Améliorer la qualité de l'image avec une méthode de “dewarping” => résultats peu probants

- Gérer la courbure des pages avec le dewarping?
- Utiliser des outils plus avancés?

PREMIER OUTIL DE DEWARPING



(a) Image d'origine
Figure - Dewarping : pas adapté à nos documents



(b) Image "dewarpee"

OUTIL DE NETTOYAGE SODUCO



(a) Image d'origine

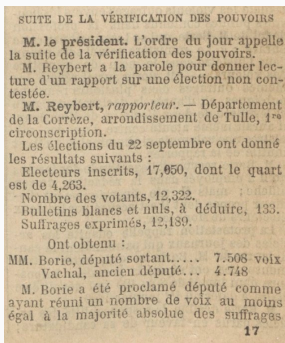
(b) Image nettoyée

Figure – Démonstration de l'outil SODUCO sur une page de débat

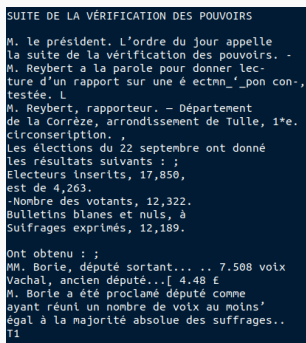
LES SOLUTIONS À NOTRE DISPOSITION

Sélectionner et comparer plusieurs moteurs OCR :

- Tesseract (ocr-tesseract ou pytesseract)
- ABBYY FineReader



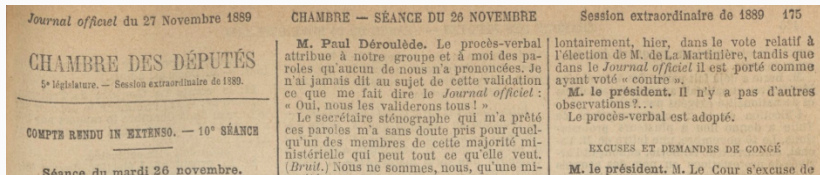
(a) Image d'origine



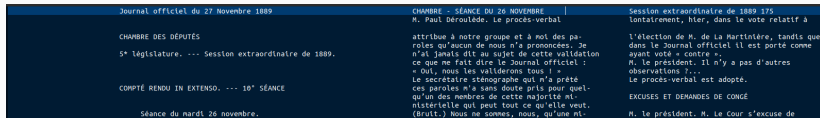
(b) OCR

Figure – Zoom sur un bloc de texte + OCR (tesseract)

LES SOLUTIONS À NOTRE DISPOSITION



(a) Image d'origine



(b) OCR

Figure – Zoom sur un bloc de texte + OCR (ABBY)

OUTIL DÉVELOPPÉ PAR LRDE (PERO OCR) - 1

Basé sur PERO OCR : très efficace sur les textes historiques

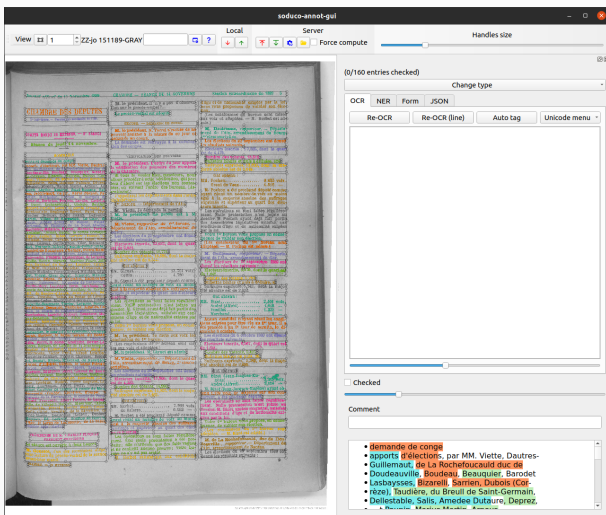
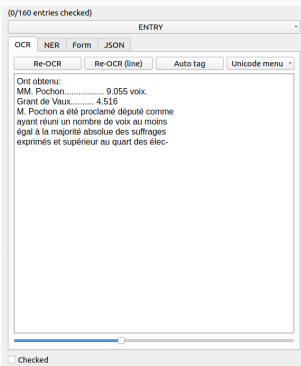
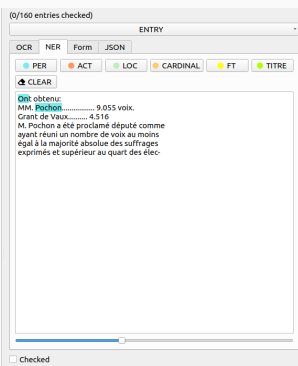


Figure – Outil LRDE

OUTIL DÉVELOPPÉ PAR LRDE (PERO OCR) - 2



(a) OCR



(b) NER

Figure – Zones d'OCR et de NER

2 Groupes de métriques :

- Méthodes non-supervisées : Dictionnaire...
- Méthodes supervisées
 - Bag Of Words : hérite de métrique classique, des limites
 - Distance de Levenshtein

Distance de Levenshtein : Nombre minimal d'insertion, délétions et substitutions sur caractères seuls pour transformer un texte en un autre. Selon les implémentations, les espaces peuvent être pris en compte ou ignorés

OCR (string token)	Ground Truth Dict	LEV _{1,1,1}
« des ates sont accomplis »	« Des actes sont accomplis »	3
« tensor »	[tenseur, trois, cube]	2

Table – Exemples de distances de Levenshtein

$$\text{CharacterErrorRate(CER)} = \frac{\text{Lev}(\text{text}_{\text{gt}}, \text{text}_{\text{ocr}})}{\text{len}(\text{text}_{\text{gt}})}$$

$$\text{CharacterAccuracy} = \max(0, 1 - \text{CER})$$

OCR	Ground Truth	LEV	CharacterAccuracy
« des ates sont accomplis »	« Des actes sont accomplis »	3	0.88
« as aboue s0 belo »	« as above so below »	3	0.82

Table – Calcul de l'Accuracy

Exemples de librairies sur python et en ligne de commande

- Jiwer : <https://pypi.org/project/jiwer>
- Fastwer : <https://pypi.org/project/fastwer>
- Outil PRImA : <https://www.primaresearch.org/tools/PerformanceEvaluation>

OCR	Document	CharAccuracy
ocr-tesseract (eng,psm3)	26 Novembre 1889	0.93
ocr-tesseract (fr,psm3)	26 Novembre 1889	0.95
ABBYY	26 Novembre 1889	0.22

Table – Résultats sur un de nos document

On remarque la limite de la distance de Lenvenshtein, elle est trop sensible à l'ordre de détection.

Une métrique moins sensible à l'ordre de détection du texte :

1. Sépare les deux textes en sacs de lignes (GT + OCR)
2. Ordonne les lignes de la GT de la plus longue à la plus petite
3. Pour la première ligne, chercher le meilleur match dans les lignes de l'OCR
4.
 - Si match complet on sort la ligne
 - Sinon on sous-divise et on rajoute à la pile de lignes
5. On recommence l'étape précédente
6. On compte le nombre de modifications nécessaires pour les lignes qu'on ne peut pas matcher
7. Calcul de l'accuracy

MÉTRIQUE : FLEXIBLE CHARACTER ACCURACY

```
ce qui est en haut  
est comme  
ce qui est en bas
```

(a) Ground Truth

```
1 est comme  
2 ce qui est en bas  
3 ce qui est en haut
```

(b) OCR

Figure – Exemple A

```
Première Ligne  
row 2  
Third line  
Last row
```

(a) Ground Truth

```
Première Ligne   Third line  
row 2           Last row
```

(b) OCR

Figure – Exemple B

Exemple	CharAccuracy	FlexCharAccuracy
A	0.45	1
B	0.39	0.95

Table – Character Accuracy vs Flexible Character Accuracy

OCR	Document	CharAccuracy	FlexCharAccuracy
ocr-tesseract (fr,psm3)	26 Novembre 1889	0.95	0.95
ABBY	26 Novembre 1889	0.22	0.97

Table – Résultats sur un de nos documents

- La flexible character accuracy est moins sensible à l'ordre de détection
- Résultats sensiblement équivalents quand l'ordre est le bon
- Comparaison de la qualité de détection des caractères toutes choses égales par ailleurs
- Utile si on n'utilise des algorithmes qui ne s'occupent pas de l'ordre des mots (Clustering via frequency matrix, topic modelling : LDA...)

- Dictionnaire de post-correction
- Utilisation d'expressions régulières : gérer les espaces multiples, passage à la ligne, les « - »
- Corrections endogènes

Malgré les progrès de l'OCR, pas de résultats parfaits :

- Fortement dépendants de la qualité de la numérisation et des documents
- Même si imprimés => phase d'entraînement en général nécessaire
- Phase de post-correction



Aurélien Pellet : aurelien.pellet@epitech.eu

Marie Puren : marie.puren@epitech.eu