

Contenu de la documentation

Présentation	3
Contexte	3
Objectifs	3
1 HTR	5
1.1 Principes généraux	5
1.2 Enjeux et tâches préliminaires	7
1.2.1 Des sources écrites par plusieurs mains	7
1.2.2 Finalité : l'édition des lettres	8
1.2.3 Choisir des collections d'évaluation et identifier des mains	8
1.2.4 L'écriture personnelle de C. de Salm : un défi paléographique	11
1.2.5 Préparer le traitement d'un dossier	11
1.2.6 Transkribus ou eScriptorium ?	14
1.3 La segmentation	16
1.3.1 Principes et enjeux	16
1.3.2 Définir une structure de document idéale en vue de l'édition	18
1.3.3 Problèmes posés par l'espacement des lignes	20
1.3.4 Gérer la numérotation des lignes	22
1.3.5 Définir une ontologie des régions et des lignes	25
1.3.6 Entraîner des modèles de segmentation des pages	27
1.3.7 Résultats des entraînements	27
1.3.8 Contrôler la pertinence de la segmentation	28
1.4 La reconnaissance des caractères	29
1.4.1 Sélectionner des échantillons d'écriture et organiser les fichiers	29
1.4.2 Établir des normes de transcription	32
1.4.3 Transcription manuelle <i>versus</i> transcription automatique	34
1.4.4 Éliminer d'une transcription les lignes attestant des écritures parasites	35
1.4.5 Comparer les performances des modèles	35
1.4.6 Tenir un journal des résultats de tests et d'entraînements	38
1.4.7 Injecter les transcriptions manuelles dans les prédictions	39

1.5	La correction semi-automatisée	39
1.5.1	Trouver le bon compromis entre granularité et performance	40
1.5.2	Analyser les mots	41
1.5.3	Gérer les résolutions ambiguës	42
1.5.4	Élaborer et enrichir un nouveau dictionnaire de la langue française	43
Annexes		44
A	Transcriptions	
	de deux manuscrits autographes	
	de C. de Salm	47
A.1	Premier extrait	47
A.2	Second extrait	49
B	Normes de transcription	51
B.1	Accentuation	51
B.2	Majuscules et minuscules	51
B.3	Séparation des mots	51
B.4	Orthographe	51
B.5	Abréviations	52
B.6	Ponctuation	52
B.7	Passages biffés, palimpsestes	52
B.8	Passages illisibles	52
Glossaire		53
Acronymes		55
Bibliographie		57

Présentation

Contexte

Constance de Salm (C. de Salm) (1767-1845), femme de lettres française, a entretenu une vaste correspondance à partir de son mariage avec de nombreux intellectuels en Allemagne, en France, en Russie.

Le projet de publier numériquement sa correspondance est né de l'intérêt pour les relations entre noblesses française et allemande au sein du Deutsches Historisches Institut Paris (DHIP). Il en a résulté la production d'un site *Wordpress* adossé au système de base de données Die Virtuelle Forschungsumgebung für die Geistes- und Sozialwissenschaften (FuD). Les notices de plus de 11000 lettres, publiées sur le site constance-de-salm.de, associent la reproduction numérique des documents manuscrits (lettres, copies, brouillons, recueils) avec leurs métadonnées descriptives, ainsi qu'une transcription de la première ligne de chaque lettre.

Objectifs

L'objectif du stage consiste à mettre en place un flux de production automatisé pour l'édition des lettres au format Text Encoding Initiative (TEI)¹. On s'appuiera pour cela sur les instruments et la documentation produits dans le cadre du projet Digital Edition of historical manuscripts (DAHN), fondé sur l'édition de la correspondance de Paul d'Estournelles de Constant (1852-1924)².

Il s'agit en particulier d'identifier les points de difficulté que posent le traitement de ce vaste corpus tant du point de vue de la transcription automatisée des documents que du point de vue de leur encodage au format TEI.

Il serait notamment souhaitable, au terme du stage de disposer d'un flux de production pour l'édition d'un volume de recueil de lettres.

1. La *Text Encoding Initiative* propose des principes pour l'encodage des textes ainsi qu'un format standard pour l'échange des données textuelles. Depuis le début des années 2000 sa syntaxe est fondée sur le langage Extensible Markup Language (XML). Cf. *TEI : Text Encoding Initiative*, URL : <https://tei-c.org/> (visité le 16/06/2022)

2. Floriane Chiffolleau, *DAHN Project*, GitHub, URL : <https://github.com/FloChiff/DAHNProject> (visité le 05/04/2022).

Chapitre 1

Reconnaissance automatique des écritures manuscrites (*Handwritten Text Recognition (HTR)*)

La reconnaissance automatique des écritures manuscrites (ou HTR) se fonde sur des principes techniques globalement similaires à la reconnaissance optique des caractères (imprimés) (ou *Optical Character Recognition (OCR)*), et il est courant de ne pas établir de distinction fondamentale entre ces deux techniques, bien que leur mise en application fasse appel à des logiciels différents (il sera question plus loin de Transkribus et eScriptorium)¹.

1.1 Principes généraux

La reconnaissance automatique des écritures manuscrites recouvre quatre phases indissociables et complémentaires :

1. L'import des images dans l'application : dans le cas présent il s'agit simplement de convertir les images stockées sur un disque dur du format non compressé **tiff** (qui permet d'archiver des images de la meilleure qualité possible) vers le format compressé **jpeg** (qui permet de travailler avec une bonne qualité d'image sous la forme de fichiers plus légers) ;
2. La segmentation des pages, au cours de laquelle les textes contenus sur chaque page sont repérés par zone et les lignes qui composent ces zones de texte sont identifiées et numérotées dans l'ordre de lecture (ce sans quoi la transcription produite serait inexploitable !) ;

1. Certaines publications tentent d'introduire une distinction entre les deux techniques dans la mesure où les techniques d'OCR se fondent souvent sur la reconnaissance caractère par caractère et non sur la reconnaissance des lignes (employée par toutes les techniques HTR), mais ce n'est pas toujours le cas (Peter A. Stokes, Benjamin Kiessling, Daniel Stökl Ben Ezra, Robin Tissot et Gargem El Hassane, « The eScriptorium VRE for Manuscript Cultures », *Classics@ Journal* (, 29 juil. 2021), URL : <https://classics-at.chs.harvard.edu/classics18-stokes-kiessling-stokl-ben-ezra-tissot-gargem/> (visité le 15/06/2022)).

3. La reconnaissance des écritures proprement dite, ou transcription automatique, qui procède à l'identification de chaque caractère sur les lignes précédemment repérées ;
4. La compilation des lignes transcrives dans un document cohérent pour chaque image traitée et l'export du résultat dans un format exploitable : on a en l'occurrence retenu le format *Analyzed Layout and Text Object* (ALTO), maintenu par la Bibliothèque du Congrès et privilégié par la Bibliothèque nationale de France (BnF)^{2 3}.

Les phases les plus délicates sont naturellement la troisième et la quatrième en ce qu'elles reposent toutes deux sur l'apprentissage machine (*machine learning*) ou appren-tissage supervisé. Cette méthode implique la constitution de données d'entraînement de façon manuelle, données qui sont ensuite analysées de manière statistique par l'outil informatique. Au terme de cette phase est produit un modèle capable, avec un taux d'acuité exprimé en pourcentage, de reproduire l'opération initialement effectuée manuellement, qu'il s'agisse de la reconnaissance des régions et des lignes d'écriture (segmentation) ou de la transcription des caractères.

Ce processus d'entraînement comporte deux phases longues et consommatrices d'énergie :

1. La constitution des données d'entraînement par l'homme : segmenter et transcrire à la main un nombre de pages suffisamment important pour un entraînement efficace ;
2. L'entraînement par la machine, qui demande une puissance de calcul très importante (selon le matériel utilisé et la quantité de données, un entraînement peut durer de quelques heures à... quelques semaines) et donc une forte consommation d'électricité.

Une fois qu'un modèle satisfaisant est produit, son utilisation est en revanche rapide et très peu consommatrice ; ainsi, plus la quantité de données pouvant être traitée par un modèle est grande, plus l'opération dans son ensemble est rentable. De plus, un modèle produit à partir de sources déterminées peut être réutilisé dans un contexte différent, et fort heureusement, le développement des projets faisant appel à la reconnaissance automatique des écritures manuscrites a engendré la multiplication des données d'entraînement et des modèles pré-entraînés. Il ne s'agit donc pas de partir de zéro mais d'abord et avant tout d'identifier les meilleurs modèles à partir desquels procéder à de nouveaux entraînements ou affinages, afin de les rendre plus adéquats aux sources sur lesquels on travaille.

2. *Techniques et formats de conversion en mode texte*, BnF - Site institutionnel, 2022, URL : <https://www.bnf.fr/fr/techniques-et-formats-de-conversion-en-mode-texte> (visité le 16/06/2022).

3. Id., « The eScriptorium VRE for Manuscript Cultures »...

Il faut aussitôt mettre un bémol à cet état de faits encourageant : les deux phases de la segmentation et de la transcription ne jouissent pas du tout des mêmes possibilités quant à la réutilisation de modèles. Si les modèles de transcription sont déjà nombreux et, lorsque les écritures ne sont pas trop cursives, peuvent être affinés de façon satisfaisante sur de nouvelles écritures avec seulement une dizaine de pages transcris à la main, il n'en va pas de même des modèles de segmentation, comme on aura l'occasion de le voir plus loin⁴.

En résumé, évaluer la rentabilité de la reconnaissance automatique des écritures manuscrites suppose avant tout d'évaluer les caractéristiques graphiques des sources d'une part (la mise en page des documents et les styles d'écritures permettent-ils d'entraînement facilement des modèles performants ?), et de définir les finalités du travail d'autre part. Il en sera question dans ce chapitre. En outre, on justifiera la sélection des sources sur lesquelles nous avons travaillé, l'intégralité de la correspondance n'ayant évidemment pu être traitée en quatre mois de stage. Enfin, on discutera du choix des applications utilisées pour procéder à la reconnaissance automatique de l'écriture.

1.2 Enjeux et tâches préliminaires

1.2.1 Des sources écrites par plusieurs mains

Quatre à cinq mains différentes ont été repérées jusqu'à présent dans la correspondance de C. de Salm, mais aucune enquête paléographique complète n'a été menée et l'on peut donc supposer une bien plus grande variété paléographique dans l'ensemble des dossiers.

Cette variété des écritures est un problème majeur pour l'automatisation des transcriptions. Les réflexions issues du projet Lecture Automatique de Réertoires (Lectaurep) ont permis de guider notre démarche. L'alternative méthodologique a été décrite ainsi par A. Chagué :

Quand on se lance dans une campagne de transcription reposant sur la reconnaissance d'écritures manuscrites, on passe généralement par une série de questions qui sont les mêmes d'un projet à l'autre. Parmi ces questions, il y a celle des modèles de transcription et de leur rapport à la variation des écritures. Doit-on entraîner un modèle pour chaque type d'écriture présent dans un corpus de documents ? Au contraire, peut-on se contenter d'entraîner un seul modèle tout terrain (qu'on appellera mixte ou générique)⁵ ?

4. Voir *infra* 1.3, p. 16.

5. Alix Chagué, *Création de modèles de transcription pour le projet LECTAUREP #1*, Lectaurep : l'intelligence artificielle appliquée aux archives notariales, URL : <https://lectaurep.hypotheses.org/475> (visité le 05/04/2022).

Les résultats probants obtenus par le projet Lectaurep en suivant l'option d'entraînement d'un modèle mixte⁶ nous ont convaincu d'emprunter cette voie. Deux séries de tests méritaient dès lors d'être effectués :

1. Reprendre les tests sur le modèle entraîné de zéro par H. Souvay lors d'un précédent stage consacré à la correspondance de C. de Salm⁷;
2. Reprendre un modèle générique entraîné dans le cadre du projet Lectaurep pour en évaluer les performances.

1.2.2 Finalité : l'édition des lettres

À la différence de l'analyse textométrique ou de l'interrogation du texte brut, finalités très courantes de la reconnaissance automatique d'écriture, l'édition ne peut tolérer que quelques fautes de transcription persistent dans la production finale. Théoriquement, le texte doit être établi à la perfection (bien que l'erreur humaine soit toujours possible). Or, la reconnaissance automatique d'écriture ne parvient jamais à une acuité de 100% : la reconnaissance des espaces et des signes de ponctuation est particulièrement problématique, et les variations paléographiques inhérentes à toute écriture manuscrite entraînent fatalement des erreurs de reconnaissance, même avec un modèle particulièrement adapté à l'écriture en question⁸.

L'évaluation des performances des modèles est donc un élément capital de cette phase du travail, car en-dessous d'une acuité estimée autour de 95%, la reprise des prédictions automatiques du texte par l'éditeur devient tellement fastidieuse que le bénéfice de la reconnaissance automatique devient caduc, imposant de procéder par une transcription manuelle. Une série de prédictions sera donnée en exemple pour apprécier l'écart entre une prédition d'une acuité voisine de 90% (insuffisante pour l'édition) et une prédition d'une acuité supérieure à 95%⁹.

1.2.3 Choisir des collections d'évaluation et identifier des mains

Afin de donner les meilleures chances à l'évaluation du modèle déjà entraîné par H. Souvay, nous sommes repartis des mêmes vérités de terrain, issues de la seconde copie de la correspondance générale.

6. Id., *Création de modèles de transcription pour le projet LECTAUREP #2*, Lectaurep : l'intelligence artificielle appliquée aux archives notariales, URL : <https://lectaurep.hypotheses.org/488> (visité le 05/04/2022).

7. Hippolyte Souvay, *La Correspondance de Constance de Salm (1767-1845) : Rapport de Stage*, rapport de stage de seconde année de master Humanités numériques et computationnelles, École nationale des chartes-Institut historique allemand à Paris, 2021.

8. Une acuité de 99% est atteignable (P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The Escriptorium VRE for Manuscript Cultures »...), mais il a semblé plus raisonnable de ne pas investir trop de temps dans l'entraînement du modèle et de se contenter des excellents résultats atteints.

9. Cf. ??, p. ?? *et passim*.

Ces recueils de lettres constituent la part du corpus la plus normée sur le plan de l'écriture et de la mise en page, leur qualité de conservation assurant en outre de bonnes conditions à la reconnaissance d'écriture. Nous avons particulièrement exploité les trois premiers volumes de cet ensemble qui en compte six¹⁰.

La variété des écritures se partage de manière contrastée entre des mains dominantes et des mains rares. Généralement, deux mains dominantes se partagent un recueil ; leur distribution peut être discontinue. Quant aux mains rares, elles n'occupent que quelques feuillets par recueil ; nous ne les avons pas retenues pour les tests, car la meilleure méthode consiste à transcrire ces pages à la main.

Trois mains principales ont pu être identifiées dans ces trois premiers volumes. La première est la plus représentée des trois.

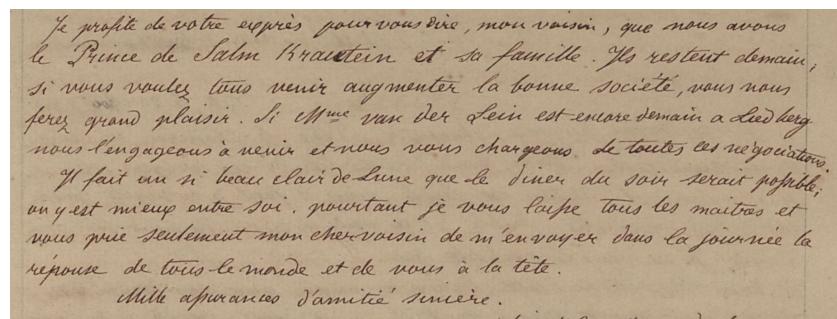


FIGURE 1.1 – Première main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-02_0065.jpg).

Une main est particulièrement attestée dans la première moitié du premier volume ; elle est ici qualifiée de « deuxième main principale ».

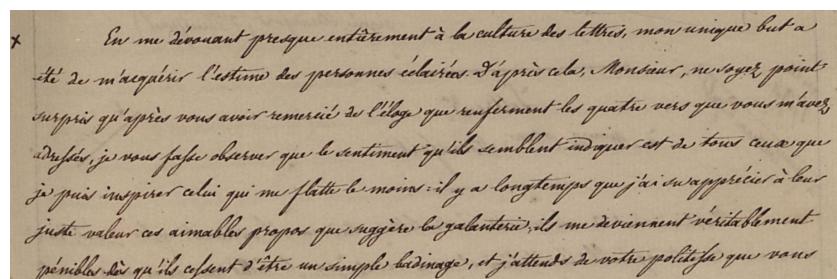


FIGURE 1.2 – Deuxième main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-01_0030.jpg).

L'écriture qualifiée de « troisième main principale » est sporadiquement attestée dans les trois volumes, mais a néanmoins été identifiée sur presque 160 pages.

10. CdS/02_1/001-330 : Correspondance générale, seconde copie, 1^{er} volume, 1785-1814, URL : <https://constance-de-salm.de/archiv/#/document/11215> (visité le 11/04/2022) ; CdS/02_2/001-369 : Correspondance générale, seconde copie, 2^e volume, 1815-1821, URL : <https://constance-de-salm.de/archiv/#/document/11216> (visité le 11/04/2022) ; CdS/02_3/001-334 : Correspondance générale, seconde copie, 3^e volume, 1822-1828, URL : <https://constance-de-salm.de/archiv/#/document/11217> (visité le 12/04/2022).

Ce sont ces pensées, qui d'ailleurs m'occupent depuis longtemps, qui m'ont décidée, comme je viens de le dire, d'abord à classer, ensuite à revoir, et enfin à faire imprimer pour moi ces correspondances, en attendant qu'elles puissent être jointes à mes Mémoires. Pour les rendre encore plus authentiques, j'ai pris aussi la résolution d'extraire de leurs bâches et de réunir les Originaux des lettres ci-dessus expliquées, et

FIGURE 1.3 – Troisième main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-01_0006.jpg).

Les écritures du recueil de la correspondance adressée par J.P.E. Martini à C. de Salm ont également été analysées afin d'élargir la variété de notre corpus de tests. Deux mains y ont été distingués.

Vous devrez recevoir aujourd'hui samedi à votre arrivée à Dyck la lettre que j'ai eu l'honneur de vous écrire. Mardi matin, j'ai passé ces jours-ci chez M^e de votre mère qui se plaignait d'un gros rhume. Je crois qu'il est question de convoquer le corps législatif pour le premier Juillet prochain; mais il s'agit de sanctionner le code criminel, ainsi que le code du commerce qui doit être terminé par Ordre pour cette époque, sans compter plusieurs autres objets qui seront soumis à la discussion, et qui pourront bien occuper deux Mois la session. Je ne puis me former aucune idée sur vos projets pendant ce

FIGURE 1.4 – Première main de la correspondance Martini (LAI, détail du cliché CdS02_Konv019_0002.jpg).

Je pars demain pour la campagne où je resterai 17 jours, à 17 lieues de Paris avec M^e et M^e La Baronne. Ils m'ont tant sollicité que j'ai à la fin consenti. D'autant plus que je n'ai ce n'aurai pour longtemps rien à faire à Paris. Mon récital ne sera point exécuté; le maître de musique de Notre Dame a fait quelques nouveaux morceaux pour le siège et il tient volontiers à les faire entendre; Je plus il en a le droit par sa place. Je vous dirai encore que si j'avais en 150 francs à dépenser ce pour mes menus plaisirs dans mon trésor, j'aurais pu aller vous voir à Dyck pour votre fête; mais mon Dieu ! il n'est pas à mon grand regret

FIGURE 1.5 – Seconde main de la correspondance Martini (notice CdS/19/036-037, URL : <https://constance-de-salm.de/archiv/#/document/10504>).

On a privilégié pour les corpus de test et d' entraînement des modèles des reproductions favorables à une bonne reconnaissance de l'écriture, évitant en particulier les problèmes de transparence qui font ressortir au recto l'encre du verso (un problème assez présent dans la correspondance Martini).

1.2.4 L'écriture personnelle de C. de Salm : un défi paléographique

Concernant l'écriture personnelle de C. de Salm, le site ne publie aucune lettre originale de sa main, mais 52 brouillons (*Entwurf*). Entraîner un modèle de reconnaissance sur cette écriture suppose un travail délicat de transcription d'une écriture particulièrement cursive.

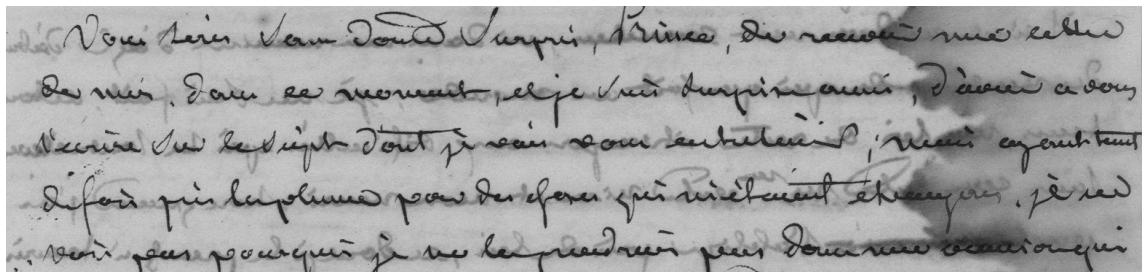


FIGURE 1.6 – Écriture autographe de C. de Salm (notice C11/S92/047-049, URL : <https://constance-de-salm.de/archiv/#/document/765> (visité le 13/06/2022), transcription *infra*, p. 49).

Nous avons tenté l'expérience de produire des vérités de terrain pour l'entraînement d'un modèle de reconnaissance propre à cette écriture, mais les résultats des premiers tests se sont révélés décourageants : la meilleure acuité obtenue ne dépassait pas 44%¹¹.

Par ailleurs, les difficultés rencontrées pour transcrire des pages de l'écriture de C. de Salm ont été importantes. Il a donc fallu renoncer à cette expérience, au risque d'y passer un temps long pour un résultat douteux.

Faute de vérités de terrain dignes de ce nom, nous donnons en annexe à ce travail les transcriptions de deux extraits de lettres¹².

1.2.5 Préparer le traitement d'un dossier : méthodologie de sélection des pièces publiées

L'archive photographique de la correspondance de C. de Salm comporte des documents non inventoriés et des éléments inventoriés mais non publiés. Un script de contrôle des données a été écrit pour dresser la liste des notices d'inventaire mais dont les données n'ont pas été validées pour la publication BIAY (Sébastien), *donneesNonPubliees.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesNonPubliees.py> (visité le 21/06/2022) ; il permet d'afficher ces données dans un fichier au format Json, en donnant la liste des images concernées.

11. Résultat obtenu rétrospectivement avec le modèle que nous avons entraîné sur quatre mains : *cds_lectcm_04_mains_01.mlmodel*.

12. Voir *infra*, A, p. 47.

À ce titre tous les dossiers ne sont pas logés à la même enseigne. Par exemple des images du premier volume de la seconde copie *CdS/02_1/001-330 : Correspondance générale, seconde copie, 1^{er} volume, 1785-1814*, URL : <https://constance-de-salm.de/archiv/#/document/11215> (visité le 11/04/2022) ont été inventoriées sans être publiées. En revanche, pour le deuxième volume *CdS/02_2/001-369 : Correspondance générale, seconde copie, 2^e volume, 1815-1821*, URL : <https://constance-de-salm.de/archiv/#/document/11216> (visité le 11/04/2022), les images non publiées n'ont pas non plus été inventoriées ; tout un lot de textes contenus dans les images de ce volume sont donc dépourvus de données d'inventaire.

Cette situation, certainement fondée sur des critères de pertinence (en particulier celui de ne pas inventorier les copies de lettres dont les originaux ont déjà leur propre notice), représente un obstacle à la gestion automatisée des transcriptions qui peuvent être constituées d'un mélange de pièces non inventoriées (et donc dépourvues de toutes données descriptives) et de pièces inventoriées (ces dernières pouvant être associées à une adresse web déjà publiée et d'autres non). Prenons l'exemple de la figure 1.7. Cette photographie contient la fin d'une lettre¹³, le début d'une autre¹⁴, et entre les deux une lettre non inventoriée, dont le titre se situe en haut de la page de droite.

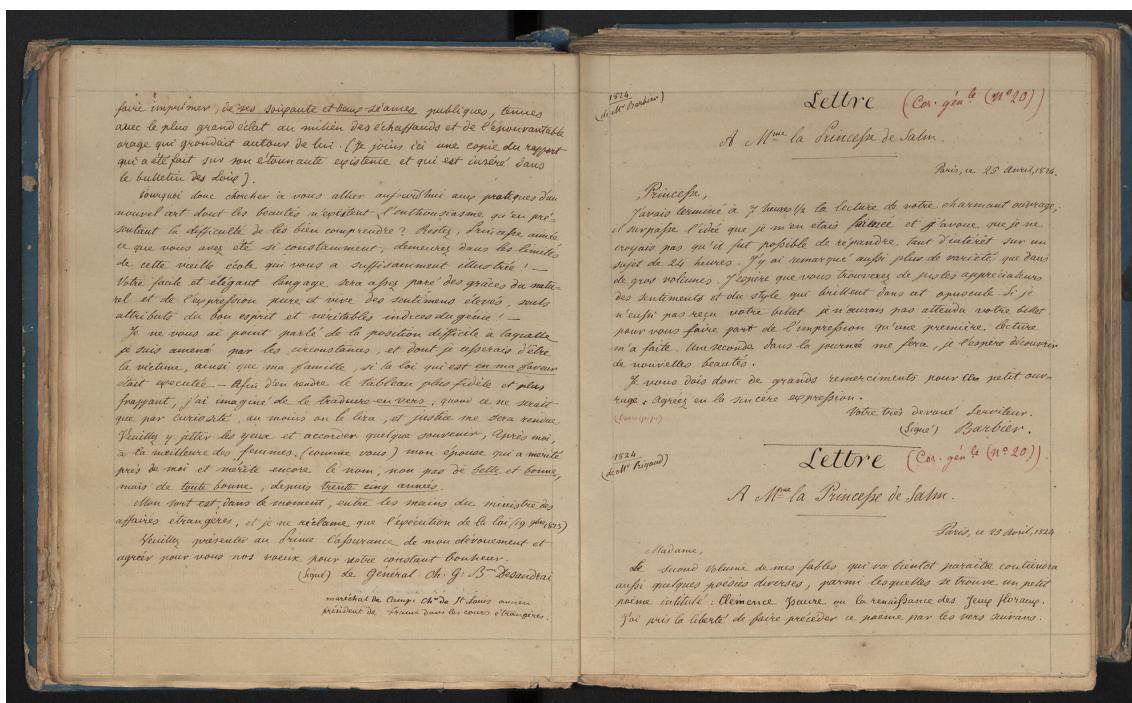


FIGURE 1.7 – Image contenant trois lettres dont l'une n'est pas inventoriée (cliché : CdS02_Konv002-03_0058.jpg)

Admettons que l'on ne souhaite éditer que les lettres inventoriées. Les données de l'inventaire permettent d'établir d'une part que la première lettre attestée dans cette

13. *CdS/02_3/057-058*, URL : <https://constance-de-salm.de/archiv/#/document/8887>.

14. *CdS/02_3/058-059*, URL : <https://constance-de-salm.de/archiv/#/document/8888>.

image commence à l'image précédente et que d'autre part l'inventaire connaît une seconde lettre dont le titre se situe dans cette image. Si l'on ne procède pas d'une manière ou d'une autre à l'élimination préalable de la lettre non inventoriée, il faudra non seulement corriger la segmentation et la transcription d'une lettre que l'on ne souhaite pas conserver dans l'encodage final (surcroît de travail inutile), mais aussi encoder à la main, pour la transcription de chaque lettre, toutes les données descriptives issues de l'inventaire, car aucun moyen de permettra alors de déterminer si le titre de la seconde lettre inventoriée sur la page est celui situé en haut de la page de droite ou s'il s'agit de celui situé en bas de la page de droite.

Face à ce problème et à la présence (inégale selon les dossiers) de données non publiées dans l'inventaire, le parti a été pris de n'engager dans notre chaîne de traitement que des documents non seulement inventoriés mais dont les notices ont en outre été publiées sur le site <http://constance-de-salm.de>.

Il va de soi que la chaîne de traitement peut être conduite sans tenir compte de cette étape de sélection des pièces. Si le parti devait être pris de transcrire l'intégralité d'un recueil sans distinction des pièces inventoriées et de celles qui ne le sont pas, il suffirait simplement de passer outre cette étape.

La sélection des pièces est donc une étape importante du début de la chaîne de traitement que l'on a élaborée dans le cadre de ce stage. Un *notebook* lui a été consacré¹⁵. Après l'étape préliminaire de l'import local et de la conversion des images au format Jpeg (afin de ne pas travailler avec le format Tiff, trop lourd), il est nécessaire d'établir la liste des images qui sont associées à une notice publiée de l'inventaire. Nous avons pour cela écrit un script python¹⁶ qui analyse les noms des fichiers convertis et importés localement, croise ces noms avec les données de l'inventaire et écrit en sortie un fichier Json qui liste (entre autres informations), pour chaque notice l'inventaire contenant l'une des images du dossier, l'URL de cette notice sur le site <https://constance-de-salm.de> et la liste complète des images attachées à cette notice¹⁷.

Une fois le dossier analysé et le fichier produit, les commandes que nous avons écrites dans le *notebook* permettent de n'importer dans le dossier de travail que les images correspondant à une notice de l'inventaire. Dans le cas spécifique illustré par la figure 1.7 d'une image contenant un mélange de pièces inventoriées et de pièces non inventoriées, c'est au stade de la segmentation que l'élimination des lettres non inventoriées est proposée (voir *infra*, p. 1.3.8).

15. Sébastien Biay, *Préparer Le Traitement d'un Dossier*, 20 mai 2022, URL : https://github.com/sbiay/CdS-edition/blob/6c4e4d4cff3101a154b9fa7e4a248e7ac87ff7ee/htr/Preparer_le_traitement_dune_source.ipynb (visité le 23/05/2022).

16. Id., *donneesImages.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesImages.py> (visité le 19/04/2022).

17. Le fichier donne par ailleurs la liste des images qui ne sont liées à aucune notice de l'inventaire, ainsi qu'une présentation des mêmes données d'association image-notice, mais cette fois par image et non par notice, et ce afin de permettre le contrôle visuel des zones de texte à transcrire (cf. *infra*, 1.3.8, p. 28)

1.2.6 Transkribus ou eScriptorium ? Fonctionnalités avancées *versus* science ouverte

Au moment du présent stage, les deux principales applications permettant de procéder à la transcription automatique des écritures manuscrites sont eScriptorium et Transkribus.

Différentes considérations peuvent conduire à opter pour l'une ou l'autre de ces applications¹⁸. Deux facteurs nous apparaissent particulièrement déterminant pour fonder un tel choix :

1. Sur le plan théorique : l'observance des principes de la science ouverte ;
2. Sur le plan pratique : les compétences d'ingénierie des personnes chargées de mener la campagne de transcription.

Considérons dans un premier temps le plan pratique.

L'écosystème applicatif Transkribus est celui qui propose le plus grand choix de services, tant pour les utilisateurs ayant des compétences d'ingénierie élevées (logiciel Expert Client) que pour les néophytes (Transkribus Lite). Conjuguées à la facilité de prise en main de Transkribus Lite, les fonctionnalités de gestion des versions de transcription offertes par Transkribus Expert Client rendent cet écosystème le mieux à même d'héberger des campagnes de transcription de grande ampleur, faisant appel à de multiples transcripteurs, voire à de la production participative (ou *crowdsourcing*).

L'application eScriptorium, à un stade de développement moins avancé¹⁹, avec une interface dotée de moins de fonctionnalités que Transkribus (gestion des versions de transcription, annotation du texte), mobilise davantage de compétences d'ingénierie. En revanche, la gratuité totale de son utilisation et surtout la culture de science ouverte portée par la communauté qui développe et utilise eScriptorium rendent cette application tout à fait adéquate aux projets impliquant un petit nombre de transcripteurs ayant une bonne culture d'ingénierie au préalable, notamment au sein d'institutions désireuses de promouvoir la science ouverte.

En effet, pour approfondir ce dernier point, la communauté active autour du développement et de l'utilisation de l'interface eScriptorium (elle-même fondée sur le logiciel libre Kraken²⁰), promeut les principes de la science ouverte de multiples manières (développement *open-source*, respects de standards des formats numériques, ouverture des

18. Nous avons assisté le 9 mai 2022 à l'atelier organisé au sein du Data-Lab de la BnF et dont le programme est détaillé dans le billet d'Olivier Jacquot, *Transkribus / eScriptorium : transcrire, annoter et éditer numériquement des documents d'archives, ateliers à la BnF*, Carnet de la recherche à la Bibliothèque nationale de France, URL : <https://bnf.hypotheses.org/12575> (visité le 10/05/2022).

19. P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The eScriptorium VRE for Manuscript Cultures »...

20. *Kraken [Documentation]*, Kraken, URL : <https://kraken.re/master/index.html> (visité le 28/04/2022).

données de modèles, de vérités de terrain, développement d'outils auxiliaires à la transcription, à la gestion de fichiers, propositions de standards d'annotation). La possibilité de réutiliser et modifier librement le code source garantit une grande pérennité d'utilisation de ces applications et donc pour les projets qui y font appel. Un projet dépendant d'un écosystème logiciel clos tel que Transkribus court en effet le risque de ne plus pouvoir être mené en cas de défaillance de cet écosystème. Un logiciel libre installé localement pourra en revanche être maintenu et réparé, et le projet de se poursuivre une fois l'écueil franchi.

L'ouverture des données (en particulier des données d'entraînement des modèles) est également décisive pour une politique de science ouverte appliquée à l'apprentissage machine. Cette technologie repose sur la constitution de données d'entraînement. Il en découle naturellement que ces données déterminent, conditionnent les résultats obtenus par les modèles entraînés (quelles images ont été choisies, quels textes ont été transcrits pour parvenir à tel résultat). Pour comprendre le fonctionnement de ces modèles et leurs performances, il faut donc disposer d'une archive des données d'entraînement ; celles-ci doivent être exposées de manière transparente, et ainsi pouvoir être critiquées, analysées ou réutilisées. Ainsi, le logiciel Kraken permet (techniquement) et la communauté eScriptorium encourage (politiquement) la publication et le partage des vérités de terrain (qui sont les véritable données brutes d'entraînement) ainsi que des modèles eux-mêmes²¹.

On peut ajouter à cette considération sur la transparence des données le haut degré de souplesse requis par les projets d'édition scientifique. Que l'on prenne en considération les spécificités des sources éditées, les critères d'édition choisis par les chercheurs ou encore les finalités de ces projets, ces derniers impliquent une multiplicité de décisions incompatible avec l'utilisation de solutions logicielles clé en main. Les besoins particuliers de la recherche sont ainsi beaucoup mieux servis par l'emploi de briques logicielles indépendantes, modulables, entre lesquelles peuvent s'échanger les données dans des standards bien établis, plutôt que par le recours à des suites logicielles performantes mais aux fonctionnalités déterminées par une communauté de développement extérieure au projet. Le risque est en effet immense de devoir reconsidérer les attendus du projets à la découverte soudaine d'une fonctionnalité manquante ou plus souvent encore de l'impossibilité de personnaliser un mode d'expression des données²².

Pour l'ensemble de ces raisons, nous avons opté pour l'utilisation d'eScriptorium et de Kraken dans le cadre de ce stage. Le flux de travail pourra sembler complexe à un utilisateur peu aguerri en matière d'ingénierie, mais en contrepartie une documentation fonctionnelle pas à pas a été rédigée grâce à la technologie du *Jupyter notebook* qui per-

21. A. Chagué, Thibault Clérice et Laurent Romary, « HTR-United : Mutualisons La Vérité de Terrain ! », dans *DHNord2021 - Publier, Partager, Réutiliser Les Données de La Recherche : Les Data Papers et Leurs Enjeux*, Lille, 2021, URL : <https://hal.archives-ouvertes.fr/hal-03398740> (visité le 15/06/2022).

22. P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The eScriptorium VRE for Manuscript Cultures »...

met en toute théorie de mener l'intégralité des tâches que l'on a experimentées avec une expertise réduite.

Dans ce genre de configuration, une assistance pourra être requise pour l'étape la plus délicate en termes d'ingénierie : l'installation des applications nécessaires à la conduite du projet, celle d'eScriptorium étant le point le plus critique et l'installation des applications en langage Python pouvant également poser quelques difficultés. Nous avons en effet utilisé eScriptorium à partir d'une installation locale²³, faisant appel aux seules ressources d'un ordinateur portable, à savoir sans serveur ni carte graphique externe²⁴. Cette méthode nous a permis de procéder à des entraînements de modèle à partir de petits volumes de vérités de terrain. Si des entraînements plus massifs s'avéraient nécessaires, il serait alors impératif de se tourner vers une infrastructure dotée de plus grandes capacités de calcul, ce que, par exemple, un partenariat entre le DHIP et le projet Consortium Reconnaissance d'Écriture Manuscrite des Matériaux Anciens (Cremma) rendrait possible.

1.3 La segmentation : reconnaissance des zones de texte et des lignes d'écriture

La segmentation est une étape indispensable à la réalisation d'une reconnaissance automatique de l'écriture manuscrite. Il s'agit de l'étape la plus problématique, car les possibilités d'entraînement de modèles de segmentation, permettant d'automatiser le processus, recquièrent beaucoup plus de données d'entraînement pour des résultats souvent médiocres, contrairement à la reconnaissance de l'écriture elle-même.

Cette section décrit les enjeux de cette opération et relate les expériences de segmentation qui ont été réalisées à partir des recueils de la correspondance de C. de Salm.

1.3.1 Principes et enjeux

Contrairement à certains logiciels d'OCR, qui procèdent directement à une reconnaissance optique caractère par caractère, tous les logiciels d'HTR fonctionnent en appliquant la reconnaissance optique au niveau de la ligne d'écriture (principe dénommé en anglais *line-wise text recognition*)²⁵. Cette technologie fonctionne sur la base d'un module d'analyse de la mise en page (*layout analysis module*) qui fonctionne indépendamment du type d'écriture rencontré, rendant ainsi possible pour une application comme Kraken de travailler sur tous les types d'écritures (alphabétiques et autres). Les logiciels Transkribus

23. La démarche est expliquée sur la page suivante : *Docker Install [Installation d'eScriptorium]*, GitLab, URL : <https://gitlab.com/scripta/escriptorium/-/wikis/docker-install> (visité le 15/06/2022).

24. L'ordinateur utilisé est doté d'un processeur 11th Gen Intel Core i7-1165G7 @ 2.80GHz × 8 et d'une mémoire vive de 15,4 GiB.

25. Id., « The eScriptorium VRE for Manuscript Cultures »...

et Kraken reconnaissent les lignes de base des écritures (*baselines*) et peuvent ainsi repérer des zones de texte où qu'elles se situent dans une page et quelque soit leur orientation.

Les capacités de l'algorithme par défaut de l'application Kraken ont été éprouvées à travers l'utilisation du logiciel eScriptorium. Les recueils de copies de lettres, présentant la particularité d'une mise en page incluant des manchettes aux lignes écrites en diagonale pour chaque lettre, représentaient un intérêt singulier.

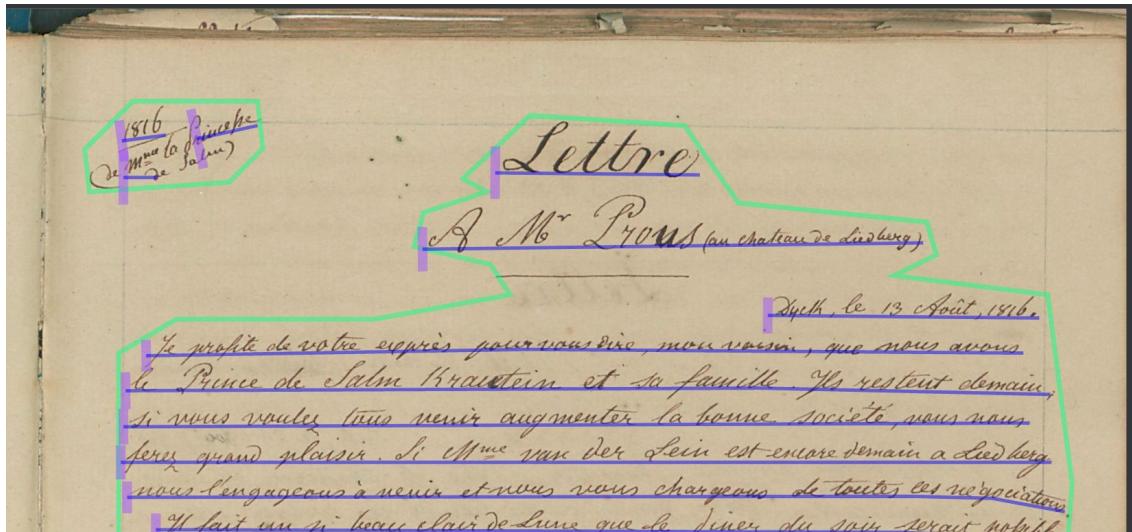


FIGURE 1.8 – Segmentation par le modèle par défaut de Kraken (LAI, détail du cliché CdS02_Konv002-02_0065.jpg).

On constate la difficulté de ce modèle standard à situer les lignes de base du texte écrit en diagonal, en haut à gauche de la page. En revanche et pour l'essentiel, le modèle a parfaitement détourné (en vert) deux régions d'écriture (la manchette et le corps de la lettre) ; il a en outre très bien repéré toutes les lignes d'écriture horizontales.

Mais si la reconnaissance correcte des lignes d'écriture garantit à priori une bonne transcription des lettres, éditer les lettres d'un tel recueil implique une autre opération : distinguer les lettres les unes des autres. En effet, et c'est l'une des caractéristiques de la mise en page de ces recueils de copie, les lettres y sont copiées à la suite les unes des autres, séparées par des titres. De plus, une lettre commence souvent sur une double page pour se terminer sur la suivante, son texte se partageant entre deux photographies. Si l'on veut éviter la tâche fastidieuse de reconstituer à la main le texte des lettres à partir des transcriptions automatiques de chaque photographie (en réunissant par copier-coller ces différentes parties dans un fichier commun), il devient crucial de structurer la transcription du texte contenu dans chaque image par le repérage dans le flux de texte d'éléments permettant cette structuration, à savoir les titres.

Rien n'est plus évident à l'œil humain que de voir un titre et de comprendre qu'il marque la fin d'une lettre et le début d'une autre. Or le modèle de segmentation par défaut de Kraken ne réalise pas cette coupure par lui-même. On le voit sur l'image suivante, la

même zone verte englobe la fin d'une lettre et le début d'une autre.

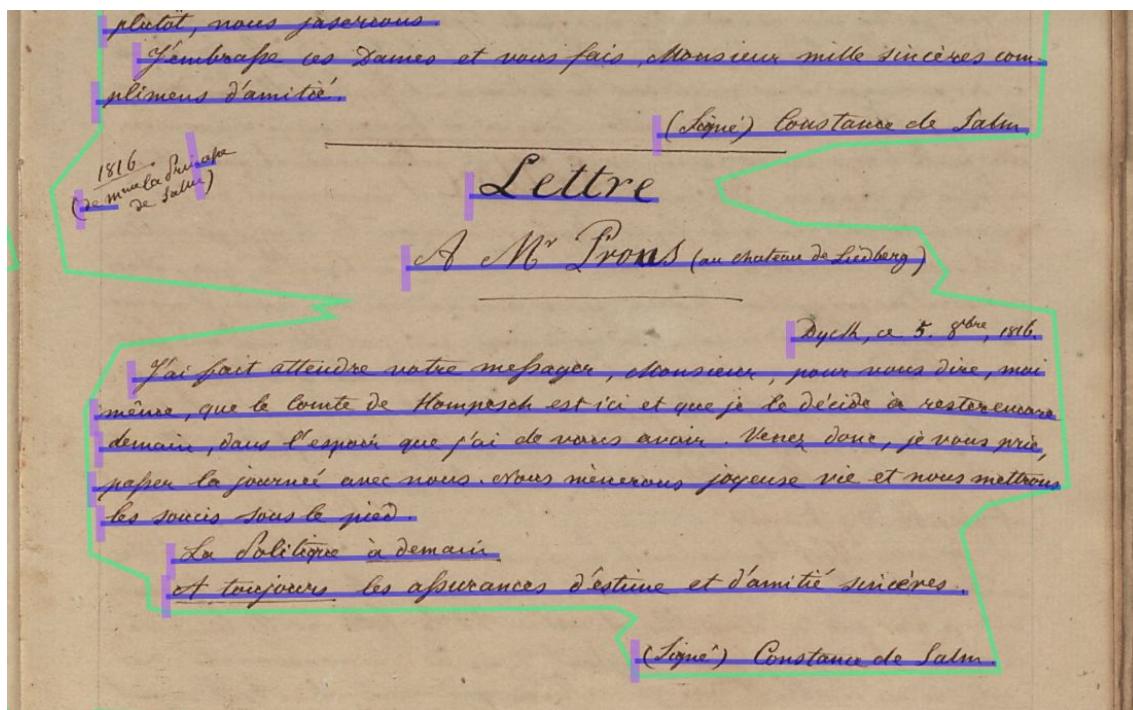


FIGURE 1.9 – Segmentation par le modèle par défaut de Kraken (notice : Cds/02_2/073, URL : <https://constance-de-salm.de/archiv/#/document/8855>).

La segmentation des pages et des lignes d'écriture pose donc une problématique. Dans quelle mesure est-il possible d'automatiser ce processus sachant qu'il est primordial que les lignes d'écriture soient correctement reconnues, qu'il est très important de pouvoir repérer les titres des lettres et que de surcroît, il n'est pas raisonnablement envisageable de créer un modèle totalement personnalisé. En effet, on a pu faire l'expérience que tenter d'entraîner un modèle à partir de zéro ne permet même pas d'aboutir à une bonne reconnaissance des lignes. La seule méthode efficace consiste donc à tirer le meilleur profit des performances du modèle standard et à tenter d'améliorer ces performances pour le rendre plus spécifique à la mise en page de ces copies de lettres.

Ces considérations générales étant posées, on se soit d'approfondir les problèmes dont les bases viennent d'être posées : le comportement du modèle standard de segmentation de Kraken à l'égard des particularités de la mise en page des sources du projet d'une part, les possibilités de structuration offertes par cette étape de la segmentation en vue de l'encodage de ces lettres au format TEI, finalité du présent travail.

1.3.2 Définir une structure de document idéale en vue de l'édition

Il est important de prendre en considération la morphologie que devra prendre l'encodage final des lettres avant même de se lancer dans l'évaluation des possibilités de

segmentation automatique des pages. En effet, l'étape de la segmentation consiste à annoter (manuellement ou, si possible, automatiquement) des zones ou régions d'écriture et les lignes qui y sont comprises. Si l'on associe, par exemple, à la ligne qui dans l'image contient le texte *Signé Constance de Salm* l'information sémantique selon laquelle il s'agit de la signature d'une lettre, il devient possible d'encoder automatiquement le texte comme étant une signature dans l'édition numérique finale.

Des propositions d'encodage extrêmement complètes pour l'édition de correspondances ont été formulées dans les *Guidelines* du projet DAHN²⁶. En lien avec ces propositions d'encodage, F. Chiffolleau a en outre élaboré une ontologie des régions d'écriture pour les correspondances en langue française pour le xx^e siècle²⁷, dans le cadre du projet SegmOnto : A Controlled Vocabulary to Describe the Layout of Pages (SegmOnto)²⁸. Voici les types de régions d'écriture dont l'application aux sources du présent projet pouvait être pertinente :

- **Main;**
- **Title;**
- **Signature;**
- **Numbering:** numbering at the top of the letter;
- **Salute;**
- **Dateline:** place and date of writing for the letter;
- **Additions:** handwritten additions outside of the main text.

Par ailleurs, les réflexions suscitées par ces catégories lors du point d'étape du 22 avril 2022²⁹ ont fait émerger l'idée d'une simplification de cette ontologie par régions autour de trois notions principales marquant l'ouverture, le corps et la fin de la lettre :

1. *Opener*;
2. *Main*;
3. *Closer*.

Il fallait en outre envisager l'annotation des éléments périphériques (annotations, annotations et systèmes de numérotation divers) :

1. *Annotations*;
2. *Numbering*.

26. F. Chiffolleau, *Correspondence : Guidelines*, DAHN Project, 10 janv. 2022, URL : <https://github.com/FloChiff/DAHNProject/blob/master/Correspondence/Guidelines/Documentation-Correspondance.pdf> (visité le 07/04/2022).

27. Id., *[Correspondance En Langue Française, XXe s.] SegmOnto*, 10 déc. 2021, URL : https://github.com/SegmOnto/examples/tree/main/sources/lettre_fr_XXe (visité le 07/04/2022).

28. Simon Gabay, Jean-Baptiste Camps, Ariane Pinche et Claire Jahan, « SegmOnto : Common Vocabulary and Practices for Analysing the Layout of Manuscripts (and More) », dans *1st International Workshop on Computational Paleography*, Lausanne, 2021, URL : <https://hal.archives-ouvertes.fr/hal-03336528> (visité le 20/04/2022).

29. On participé à cette discussion Anne Baillot, Floriane Chiffolleau, Pauline Spychala, Evan Vire-vialle et moi-même.

Restait à savoir (et ça n'était pas la moindre question) si un modèle de segmentation serait capable d'identifier ces régions d'écritures de manière automatique.

1.3.3 Problèmes posés par l'espacement des lignes

La mise en page des lettres dans les recueils de copie répond à des principes clairs pour l'œil humain ; il présente en revanche d'importantes variations métriques dans l'espace de la page. On a déjà pointé précédemment un problème de taille : les lettres sont écrites les unes à la suite des autres, et le modèle de segmentation par défaut de Kraken ne se montre pas capable de les séparer de lui-même.

Ce modèle procède à l'évaluation de l'espacement entre les régions d'écritures : il fusionne dans une seule région les lignes qui lui semblent suffisamment proches ; il assigne en revanche à des régions distinctes les lignes qui lui semblent suffisamment éloignées. On a vu que cette appréciation lui permet de distinguer la manchette des lettres, caractéristique de ces recueils de copie (voir *supra*, figure 1.8, p. 17). L'exemple illustré par la figure 1.10 montre la capacité de ce modèle à distinguer l'ouverture de la lettre (*opener*) en réunissant dans une région cohérente et indépendante le titre de la lettre, le lieu et la date (ainsi qu'une annotation de type catalographique écrite en rouge).

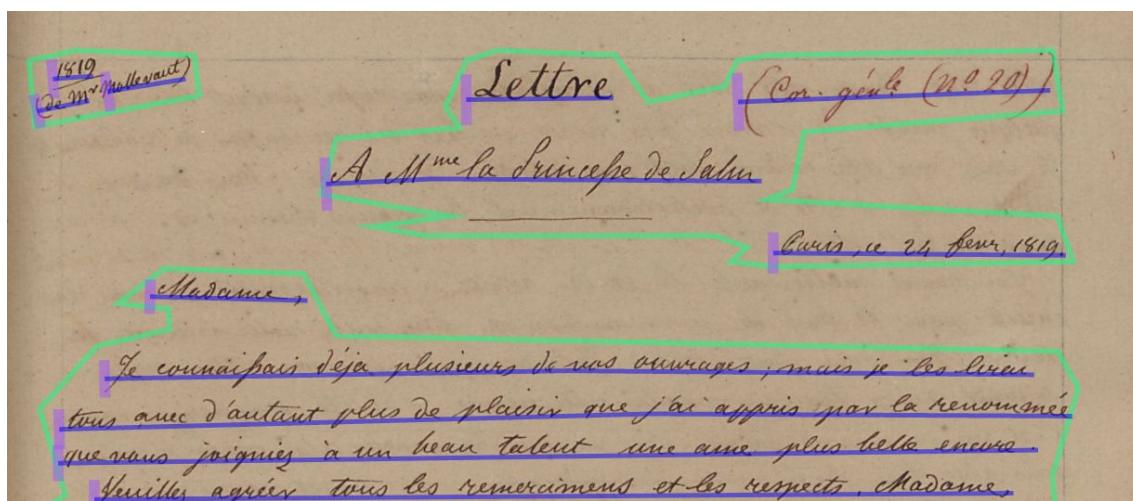


FIGURE 1.10 – Mise en page aérée au début d'une lettre (LAI, détail du cliché CdS02_Konv002-02_0193.jpg).

En revanche, dans l'exemple illustré par la figure 1.11, les lignes d'écriture sont tellement peu espacées sur l'axe vertical qu'une seule région de texte réunit le titre, la date et le corps de la lettre.

La reconnaissance de la fin d'une lettre (dont la signature alignée à droite de la page est l'élément le plus récurrent) est encore plus délicate, car elle ne se manifeste jamais par un élément visuellement massif comme un titre. De plus, les mêmes différences de comportement s'observent selon l'espacement des lignes : dans quelques cas rares, comme

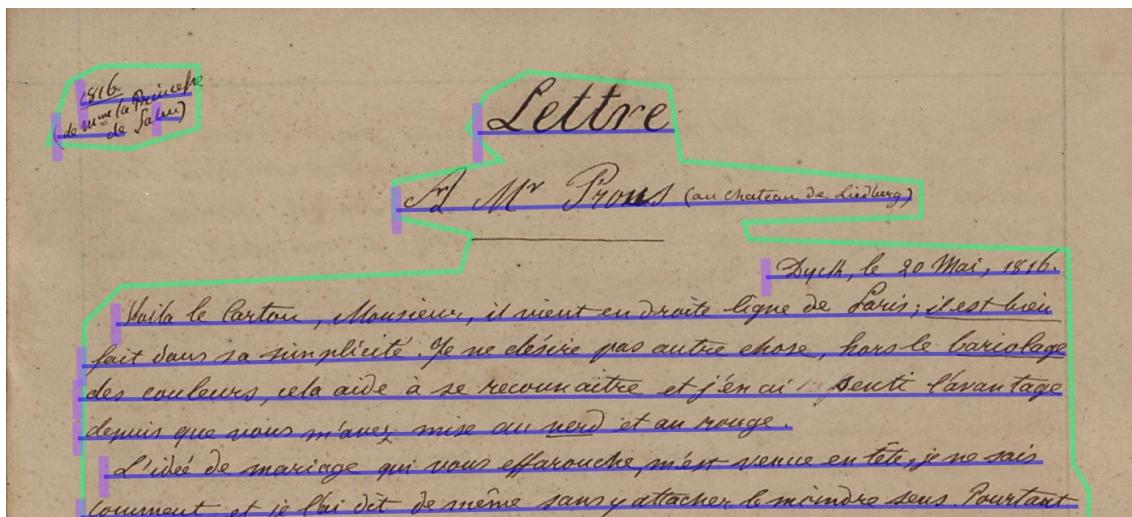


FIGURE 1.11 – Mise en page resserrée au début d'une lettre (LAI, détail du cliché CdS02_Konv002-02_0046.jpg).

celui illustré par la figure 1.12, la signature peut être distinguée en une région spécifique, mais dans la très grande majorité des cas, elle est perçue comme appartenant à la même région que le corps de la lettre.

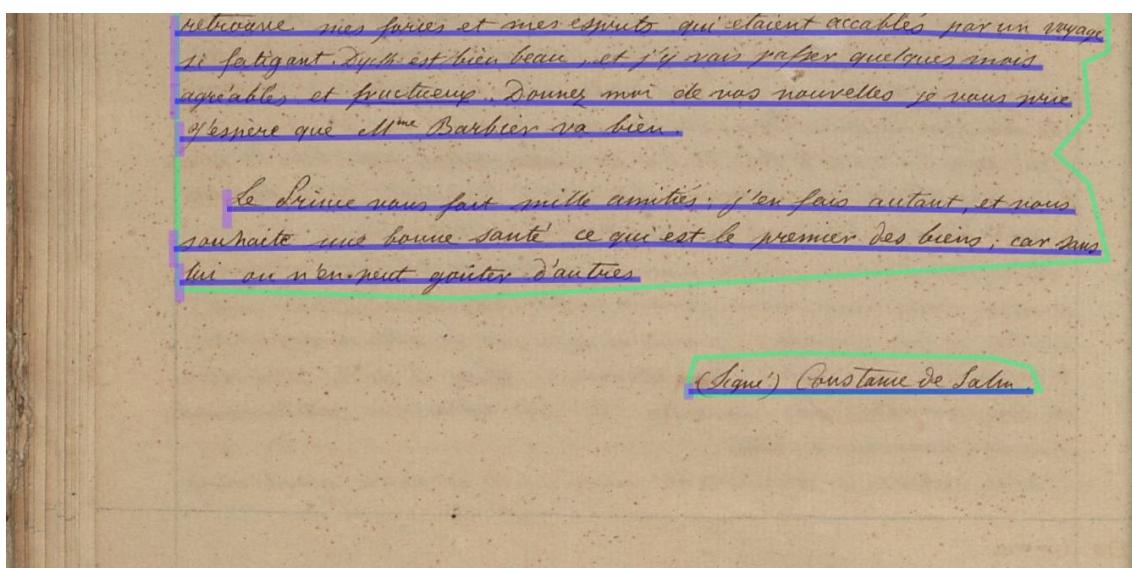


FIGURE 1.12 – Mise en page aérée en fin de lettre (notice : CdS/02_3/070-071, URL : <https://constance-de-salm.de/archiv/#/document/8907>).

Ces premières observations suggèrent qu'entraîner un modèle à reconnaître les différentes parties d'une lettre sous forme de régions d'écritures distinctes (catégorisées ci-dessus *opener*, *main* et *closer*) s'annonce difficile. D'autres observations relatives à la numérotation des lignes ont confirmé les inconvénients de cette méthode.

1.3.4 Gérer la numérotation des lignes

La numérotation des lignes de texte est une opération essentielle de la segmentation. Rien ne pourrait être fait de la transcription automatique du texte si les lignes n'étaient pas transcris dans l'ordre de lecture. L'algorithme de segmentation par défaut de Kraken est bien entendu capable de numérotter les lignes selon les préférences de l'utilisateur, et dans le cas présent, de gauche à droite et de haut en bas.

Paramètre important : la numérotation des lignes dépend de la position des régions qui les englobent dans l'image analysée. Ainsi, la mise en page particulière des documents et le découpage du texte par régions peut jouer un rôle déterminant dans l'ordre des lignes. L'expérience a pu en être faite en remodelant de différentes manières les segmentations automatiquement produites le modèle par défaut de Kraken.

Or les problèmes de numérotation ont été nombreux lors de ces tests. Comme le montre la figure 1.13, issue du vol. 2 de la seconde copie³⁰, obtenir une bonne numérotation des lignes de titre s'est révélé problématique dès lors que l'on souhaitait isoler ces titres dans une région propre (*opener*). On observe en effet dans cet exemple que le titre situé sur la page de gauche est numéroté 23 et 24 ; il ne s'intercale donc pas correctement entre les lignes de la manchette (10 et 11) et les lignes du corps de la lettre (à partir de 12). Le même phénomène s'est produit dans cet exemple pour les trois titres de la double page. Sachant que le titre est le seul élément sur lequel on puisse s'appuyer pour distinguer automatiquement le début et la fin des lettres (car il n'y a pas toujours de signature, pas toujours de manchette, pas toujours de date pour borner le texte des lettres), la mauvaise numérotation des lignes de titre est un énorme problème en vue de l'exploitation des prédictions : sans intervention, les titres de chaque lettre de cette page se retrouveraient disposer à la fin du texte de leur lettre et ainsi passer pour le titre de la lettre suivante...

Il est naturellement possible de corriger ces problèmes de numérotation de façon manuelle dans l'interface eScriptorium, mais il s'agit d'un travail fastidieux (chaque ligne doit être déplacée une par une) et hasardeux (le recharge de la page peut rétablir la numérotation d'origine.) Il fallait donc prévenir ce problème.

Une première solution a consisté à envelopper toutes les régions d'écriture dans une sorte de super-région, dessinée autour des autres régions précédemment définie : la page. En liant les lignes d'écriture à la page qui les englobe, l'ordre de lecture de gauche à droite et de haut en bas assure une bonne numérotation des lignes : on constate que les titres ont retrouvé leur juste numérotation.

Cette solution pouvait sembler prometteuse jusqu'au moment où il s'est agit d'examiner les fichiers ALTO compilés par le logiciel eScriptorium, où la transcription des textes de la page se trouve structurée en fonction des régions d'écriture que l'on a définies. Le choix de lier les lignes d'écritures à la super-région *page* les avait (logiquement)

30. CdS/02_2/001-369 : Correspondance générale, seconde copie, 2^e volume, 1815-1821...

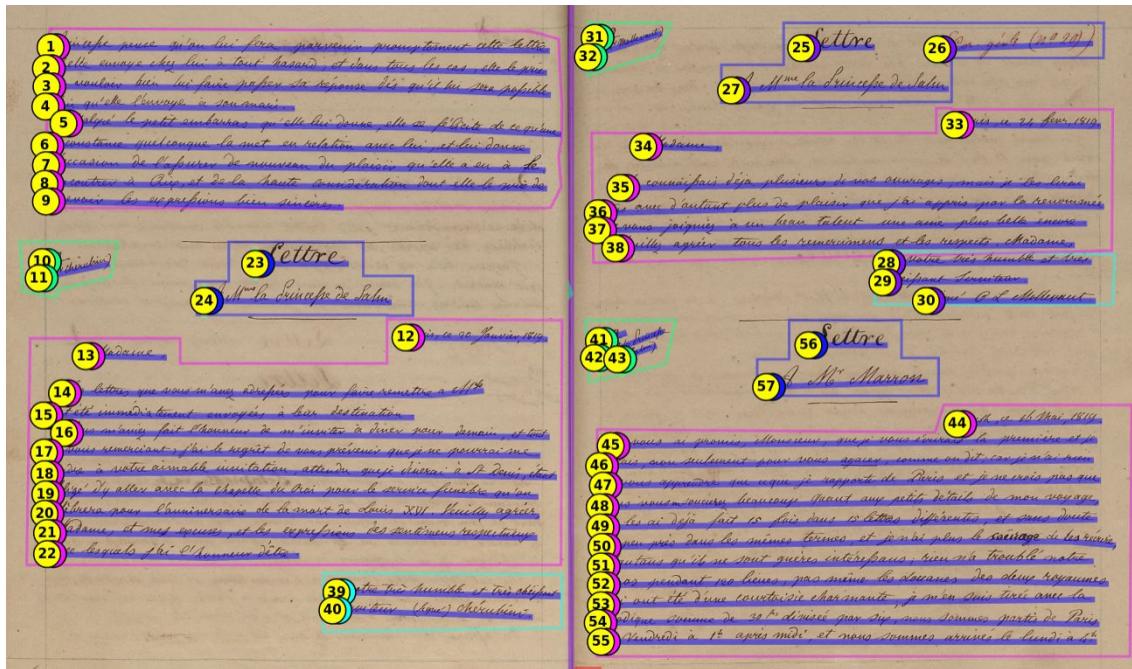


FIGURE 1.13 – Exemple de numérotation à partir d'une segmentation distinguant *opener* et *closer* (cliché CdS02_Konv002-02_0193.jpg).

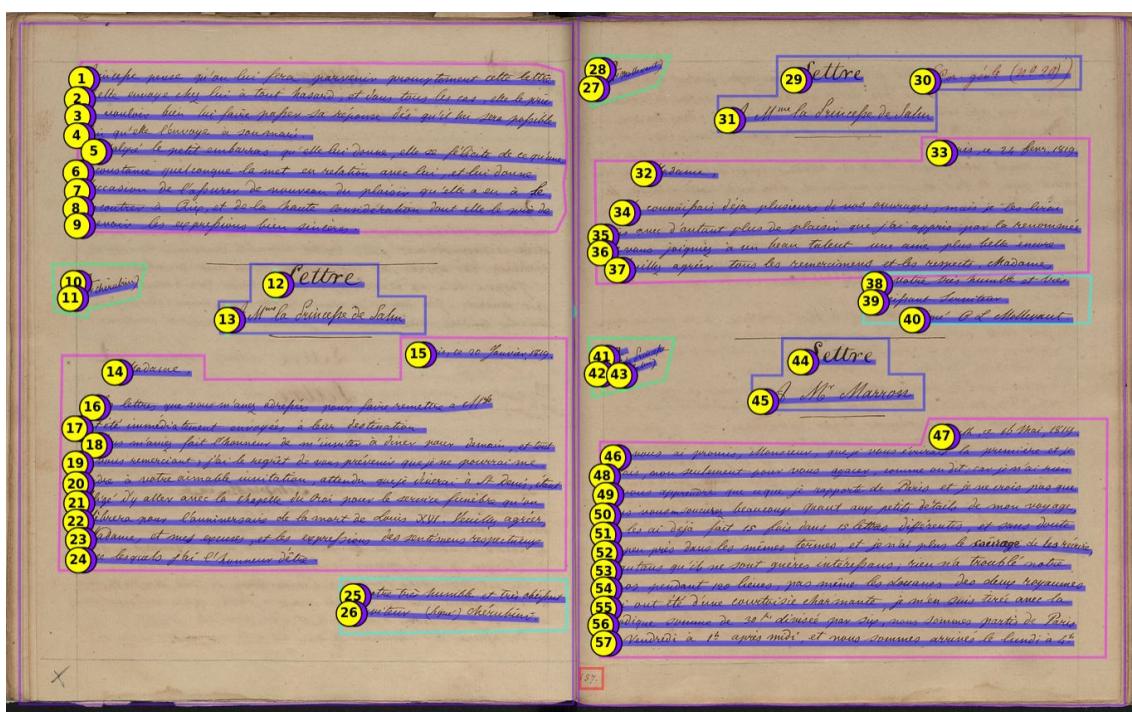


FIGURE 1.14 – Exemple de numérotation avec lignes liées à une région *page* (cliché CdS02_Konv002-02_0193.jpg).

déconnectées de leurs propres régions : le fichier se présentait avec des régions *page* pleines de lignes d'écriture et des régions *opener*, *main*, etc. entièrement vides. Il devenait dès lors impossible d'exploiter les étiquettes données à ces régions d'écriture en vue de la réalisation de l'encodage des lettres : les lettres ne pouvaient plus être distinguées les unes

des autres. Il a donc fallu renoncer à emboîter des régions d'écriture pour résoudre les problèmes de numérotation.

La solution finalement apparue comme la seule possible consistait à simplifier la définition des régions d'écriture. Dans la figure 1.15, la segmentation consiste à séparer les lettres les unes des autres et à distinguer la manchette (un choix que l'on a maintenu dans la mesure où le modèle par défaut de Kraken s'est montré capable d'opérer seul cette distinction). Toutes les parties de la lettre sont réunies dans une même région (sauf si la lettre se prolonge sur l'image suivante, naturellement). Cette option a non seulement permis de résoudre les problèmes de numérotation des titres, mais aussi de résoudre un autre problème de numérotation assez récurrent : la ligne indiquant le lieu et la date de la lettre (numéros 14, 32 et 46 dans l'exemple en question) s'intercalait assez souvent entre la première et la seconde ligne du texte de la lettre. On a également observé dans un certain nombre de cas les deux premières lignes du texte numérotées en ordre inverse.

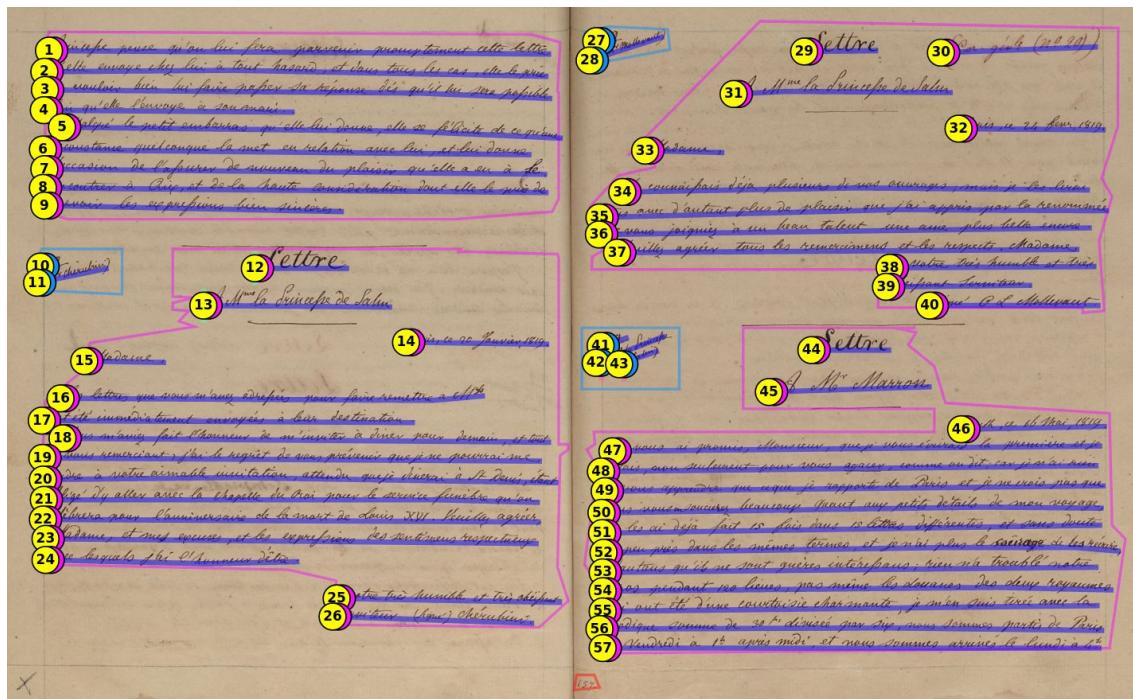


FIGURE 1.15 – Exemple de numérotation sans région de *page* ni *opener* (cliché CdS02_Konv002-02_0193.jpg).

On le voit dans l'exemple en question, il a également été choisi de ne pas distinguer de région *closer* pour la signature des lettres. Après quelques tentatives d'entraînement de modèle de segmentation en ce sens, les résultats très mauvais que l'on a obtenus prouvaient avec assez de force que cette tentative était vouée à l'échec.

1.3.5 Définir une ontologie des régions et des lignes

Une fois arrêtés les principes généraux de la segmentation, il convient de définir une ontologie, à savoir la liste des types de régions et de lignes d'écriture que l'on entend appliquer à cette segmentation.

Le choix a été fait d'inscrire cette définition des régions et des lignes suivant les principes de l'ontologie SegmOnto. Déjà cité, le projet SegmOnto propose un cadre conceptuel général pour ce genre de définition. Bien qu'orienté sur la description des manuscrits médiévaux et des premiers imprimés, il met en avant les catégories les plus génériques possibles (*main*, *margin*, *numbering*, etc.) ainsi que des solutions de personnalisation, afin de permettre à tout type de projet d'exprimer ses besoins d'annotation de régions et de lignes d'écriture dans un cadre commun. Ce cadre facilite *in fine* la réutilisation des vérités de terrain et contribue ainsi à l'ouverture des données d'entraînements des projets de reconnaissance automatique d'écriture.

Inscrire le présent projet dans ce cadre communautaire est également bénéfique au projet lui-même. La segmentation de nombreuses pages de texte, même avec l'assistance d'un bon modèle de segmentation automatique, suppose un contrôle et parfois une correction attentive des pages segmentées. Là où l'algorithme ne parvient pas à identifier un type de région ou un type de ligne, il la laisse sans annotation (type *None*). L'outil de validation d'annotation HTRUC³¹, développé selon les catégories de SegmOnto, permet de contrôler de manière très efficace que des lignes ou des régions ne sont pas restées sans annotation ; il permet encore de s'assurer que les principes de nommage des régions et des lignes ont été respectés, que ce soit pour les catégories génériques (que l'on peut mal orthographier en préparant sa segmentation) ou pour les catégories personnalisées (qui doivent être regroupées sous les catégories *CustomZone* et *CustomLine*).

En résumé, les types de régions d'écriture que l'on a retenus et conformés à SegmOnto sont les suivants :

- *CustomZone:header* pour les manchettes des recueils de copies ;
- *MainZone* pour le corps de la lettre, regroupant toutes ses parties, du titre à la signature et au post-scriptum ;
- *NumberingZone* pour tous les types de numérotation portés sur la page (pagination, numérotation des pièces), qu'ils aient été établis dès la première rédaction où qu'ils aient été ajoutés plus tard (jusqu'aux ajouts des érudits) ;
- *MarginTextZone* pour tous les types d'annotation portés sur la page (avec la même indiscrimination que pour les systèmes de numérotation) ;
- *RunningTitleZone* pour les titres courants attestés dans certains recueils comme la correspondance Martini (cf. *CdS/19/054-056*, URL: <https://constance-de-salm.de/archiv/> (visited on 06/21/2022) en haut de la page de gauche).

31. T. Clérice, *HTRUC*, *HTR-United Catalog Tooling* (Pronounced *EuchTruc*), version 0.0.1, nov. 2021, URL : <https://github.com/HTR-United/HTRUC> (visité le 20/05/2022).

On recommande l'usage de la région *MarginTextZone* pour les annotations portées sur une page qui ne seraient pas spécifiquement attachées à une lettre, car si l'on souhaite préserver le lien entre une annotation et une lettre spécifique, définir cette annotation comme une région d'écriture à part est de nature à poser de nouveaux problèmes de numérotation. On peut constater sur la figure 1.16 que la mention entre parenthèses *Correspondance gén(éra)le (n° 20)* que l'on ici isolée dans une région à part a reçu le numéro 57, ce qui l'inscrit en queue de toute la numérotation de la page. La ligne ne s'inscrit dès lors plus dans la numérotation de la lettre à laquelle on souhaiterait pourtant attacher l'annotation.

En conclusion, les annotations dont on souhaite maintenir le lien à une lettre particulière doivent être incluses dans la (ou l'une des) région(s) *MainZone* de leur lettre (ce que le modèle de segmentation par défaut effectue généralement de lui-même), le statut d'annotation étant signalé par le type de ligne qu'on lui attribue, et non par le type de région dans lequel on l'inscrit.

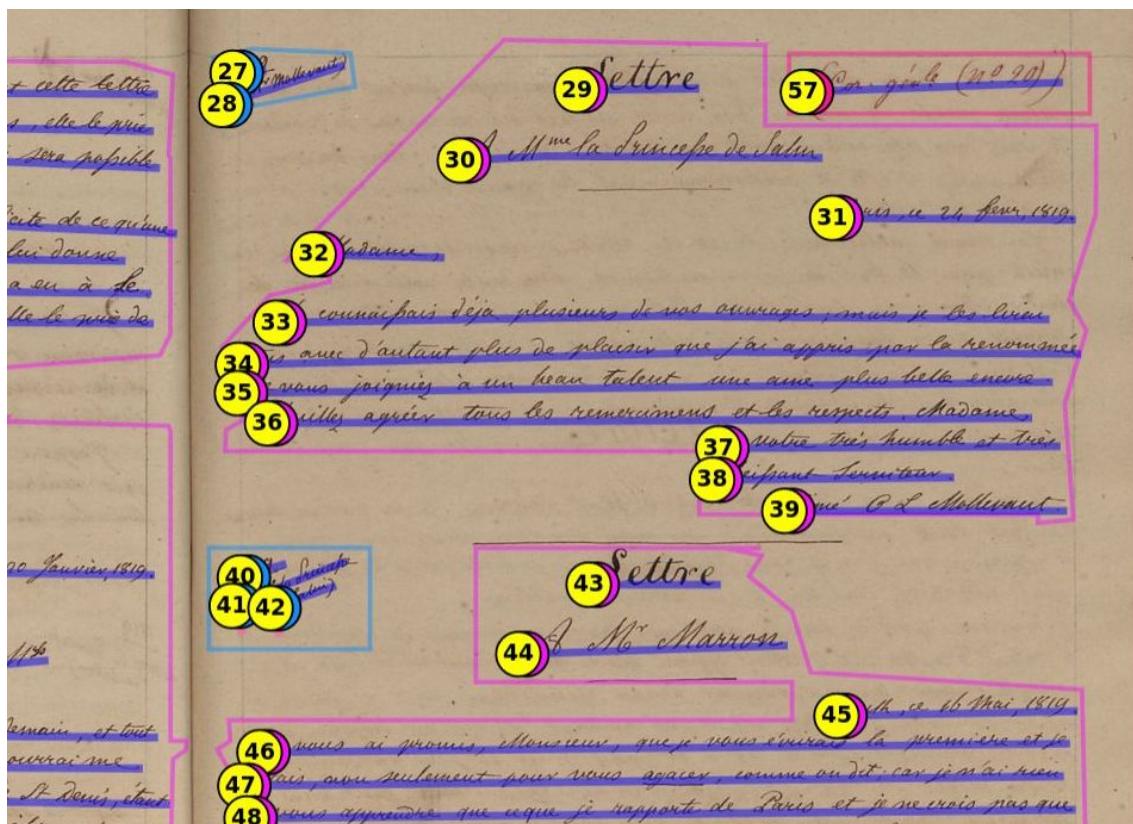


FIGURE 1.16 – Problème de numérotation lié à une région de type *MarginTextZone* (cli-
ché : CdS02_Konv002-02_0193.jpg).

Face à l'éventail limité des types de régions que l'on est contraint d'adopter pour éviter les problèmes de numérotation des lignes, l'annotation sémantique des parties de la lettre (qui permettra une automatisation partielle de l'encodage final en TEI) doit être répercutée sur les types de lignes. Voici la liste de conforme à SegmOnto que l'on propose

d'adopter, par ordre général d'apparition dans les lettres :

- *CustomLine:header* pour les petites lignes des manchettes ;
- *HeadingLine:title* pour les lignes de titre ;
- *CustomLine:dateline* pour le lieu et la date d'écriture ;
- *CustomLine:salute* pour l'éventuelle salutation initiale ;
- *DefaultLine* pour les lignes du corps de texte ;
- *CustomLine:verse* pour les éventuelles parties écrites en vers ;
- *InterlinearLine* pour les éventuelles corrections interlinéaires ;
- *CustomLine:signature* pour la signature ;
- *CustomLine:annotations* pour tous les types d'annotation.

1.3.6 Entraîner des modèles de segmentation des pages

Cette section est à écrire.

Constituer des données d'entraînement peut faire appel à deux méthodes principales : la méthode progressive ou la méthode récursive.

On entend par méthode progressive le simple fait de fabriquer à la main ses données d'entraînement de A à Z. La méthode récursive fait quant à elle appel à l'outil informatique et ce en suivant plusieurs étapes (par exemple pour un modèle dévolu à la transcription, mais la démarche serait la même pour un modèle de segmentation) :

1. Segmenter à la main quelques pages ;
2. Entraîner un premier modèle ;
3. Effectuer une segmentation automatique sur quelques autres pages ;
4. Corriger cette segmentation ;
5. Entraîner un second modèle, etc.³²

On a fait appel à la méthode récursive pour l'entraînement des modèles de segmentation ; en revanche la transcription manuelle a été privilégiée pour l'entraînement des modèles de transcription, comme expliqué plus loin³³.

1.3.7 Résultats des entraînements

Cette section est à écrire.

32. P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The eScriptorium VRE for Manuscript Cultures »...

33. Voir *infra* 1.4.3, p. 34.

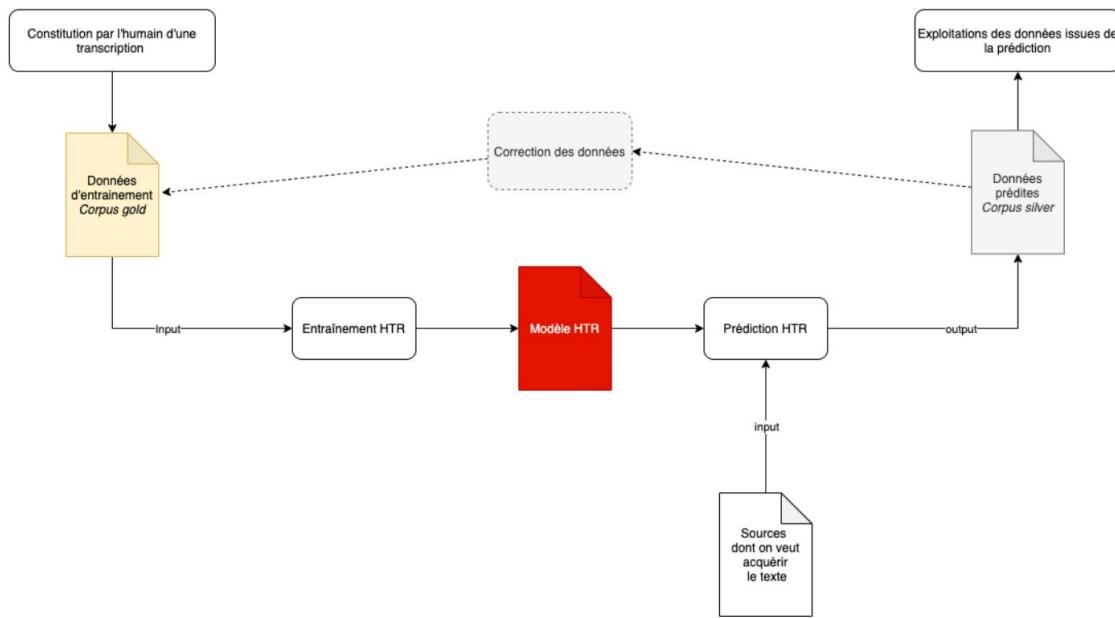


FIGURE 1.17 – Schéma d’entraînement récursif appliquée à la transcription (PINCHE (Ariane), « L’HTR : Présentation Des Problématiques Qui s’ouvrent Au Chercheur, Entre Particularité Du Document et Généralisation Du Modèle », dans *Conduite et Réalisation d’un Projet Informatique*, Cours de Master, Paris, École nationale des chartes, 2021).

1.3.8 Contrôler la pertinence de la segmentation

Le script python écrit pour permettre l’importation sélective des images inventoriées³⁴ permet en outre de contrôler l’association entre les images sélectionnées et les notices de l’inventaire. Plusieurs lettres peuvent en effet être inventoriées dans la même image, mais surtout une image peut contenir un mélange de lettres inventoriées et de lettres non inventoriées.

Afin de faciliter ce travail de contrôle, le script en question délivre pour chaque image les informations nécessaires : le nombre de lettres inventoriées dans l’image (qui permet de contrôler rapidement, en comptant le nombre de titres, si certaines parties de l’image seraient à exclure), ainsi que des informations détaillées sur chaque notice de l’inventaire concerné, dans le but de permettre un contrôle précis en cas d’ambiguïté possible. Par exemple, le cas s’est présenté d’une image contenant quatre lettres dont une seule est inventoriée ; dans ce cas heureusement rare, c’est la récupération de l’incipit de chaque lettre inventoriée par le script qui permet de repérer précisément dans l’image la ou les lettres pertinentes.

34. S. Biay, *donneesImages.Py...*

1.4 La reconnaissance des caractères

Comme énoncé plus haut, les résultats probants obtenus par le projet Lectaurep en suivant l'option d'entraînement d'un modèle mixte pour l'ensemble des écritures (plutôt qu'une série de modèles propres à une seule main) ont orienté notre démarche³⁵.

Les caractéristiques paléographiques des recueils de correspondance traités à l'occasion de ce stage apportaient un argument supplémentaire en ce sens. Les dossiers qui constituent l'archive de la correspondance de C. de Salm réunissent des documents écrits par plusieurs mains. Dans les cas les plus fréquents, chaque écriture est attestée sur une partie cohérente de recueil. Mais on a également pu constater que certaines écritures sont attestées de manière sporadique, en particulier dans les recueils de copies³⁶. Il était dès lors impossible d'envisager entraîner des modèles particuliers pour chaque écriture en découplant les dossiers par grandes zones.

Aucun modèle de reconnaissance d'écriture préexistant ne permettait d'atteindre une acuité satisfaisante sur aucune des écritures que l'on a pu identifier. La reconnaissance automatique de l'écriture supposait donc la mise en place d'une méthodologie d'entraînement d'un modèle multiple, dont le *notebook* intitulé *Tester et entraîner un modèle de reconnaissance d'écriture* explique la marche à suivre³⁷.

1.4.1 Sélectionner des échantillons d'écriture et organiser les fichiers

Entraîner des modèles à reconnaître les écritures de plusieurs mains différentes suppose un regard attentif aux variations paléographiques, mais aussi une grande rigueur de gestion des fichiers et de leurs données, car il s'agit d'abord de classer par type d'écriture les reproductions photographiques d'un dossier de la correspondance. Il est en effet essentiel de pouvoir tester les performances de modèles sur chaque main de manière isolée, afin de cibler les écritures pour lesquelles des données d'entraînements (des transcriptions manuelles) doivent être apportées. Apporter des données d'entraînements pour une main qui serait déjà reconnue par un modèle avec plus de 95% d'acuité ne serait qu'une perte de temps.

35. A. Chagué, *Création de modèles de transcription pour le projet LECTAUREP #2...*

36. C'est tout particulièrement le cas de la main dénommée mainCdS02_Konv002_03, sporadiquement attestée dans plusieurs recueils de la seconde copie des lettres ; la reproduction photographique d'un échantillon de cet écriture ainsi que la liste des fichiers où elle a pu être identifiée se trouvent sur la page *Mains* du dépôt du projet (S. Biay, *Mains*, Éditer la correspondance de Constance de Salm (1767-1845), 10 juin 2022, URL : <https://github.com/sbiay/CdS-edition/tree/main/htr/mains> (visité le 10/06/2022)).

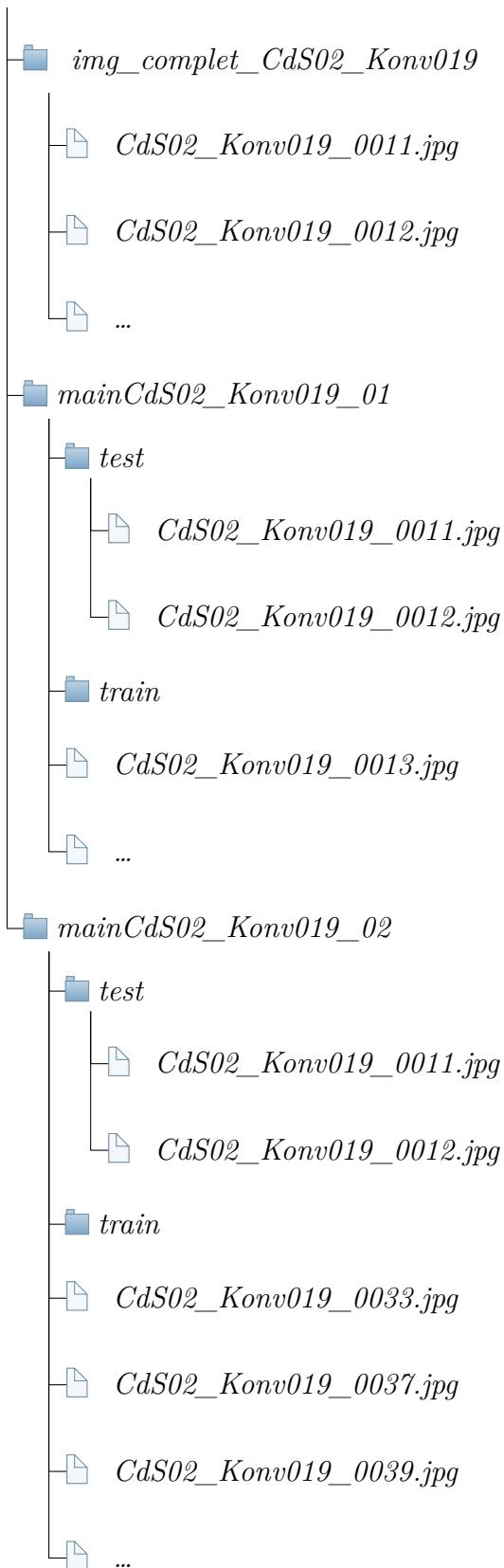
37. Id., *Tester et Entrainer Un Modèle de Reconnaissance d'écriture*, 20 mai 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/htr/Tester_et_entrainer_un_modele_HTR_avec_Kraken.ipynb (visité le 10/06/2022). Une partie de cette méthodologie a été présentée dans le cadre de la réunion mensuelle du DHIP : S. Biay et Pauline Spychala, « L'intelligence Artificielle à l'IHA », dans *Hausinfo*, Paris, IHA, 2022.

Une fois les reproductions photographiques classées par mains, il s'agit de sélectionner, pour chaque main, des échantillons pour réaliser des tests de performance de modèles de reconnaissance d'une part et des échantillons pour réaliser d'éventuels entraînements des mêmes modèles d'autre part.

Un point d'attention doit être porté à la distinction des échantillons de test et des échantillons d'entraînement. Il est en effet important que l'entraînement du modèle ne porte pas sur les mêmes échantillons que le test final de performance, car il ne s'agit pas d'évaluer la capacité du modèle à transcrire un texte qu'il aura déjà transcrit une première fois au cours de la phase d'entraînement, mais bien d'évaluer sa capacité à transcrire des textes qu'il n'aura pas encore croisés. Il est donc nécessaire de ne jamais insérer dans un échantillon d'entraînement une transcription qui servira plus tard à évaluer les bénéfices de cet entraînement.

Une méthode de nommage et de classement des fichiers a ainsi été établie afin d'uniformiser les noms et les emplacements des échantillons de test et d'entraînement (voir le schéma suivant). Ce classement permet d'une part de cibler les échantillons de manière efficace lorsqu'il s'agit de procéder à un test ou à un entraînement ; il permet d'autre part de faire analyser les dossiers de fichiers pour collecter des données sur ces mêmes opérations, comme on le verra plus loin³⁸.

38. Cf. *infra* 1.4.6, p. 38.

entrainements

Même si une image peut attester plusieurs écritures, on a retenu l'option de ne pas dupliquer l'image en question dans plusieurs dossiers de mains. En effet, les transcriptions produites à l'occasion des tests et des entraînements ont vocation à constituer une **vérité de terrain** unique : une fois ces transcriptions effectuées, elles sont ainsi rassemblées dans un seul dossier réunissant toutes les écritures (la distinction des mains n'ayant pas d'intérêt en dehors du cadre strict des tests et des entraînements). Or, si l'on transcrivait différents passages d'une même reproduction photographique pour tester ou entraîner un modèle sur plusieurs mains à partir de la même image (qu'il aura d'abord fallu dupliquer en plusieurs dossiers de mains), la réunion des fichiers dupliqués dans un dossier commun aura pour effet d'écraser les transcriptions d'une main par l'autre. Un script a donc été dédié à la vérification que l'on n'avait pas dupliqué par inadvertance un fichier dans plusieurs dossiers de mains, prévenant ainsi le risque de conflit entre les transcriptions manuelles. Il eut été également possible de prévoir la réunion des transcriptions de ces éventuels doublons en un seul fichier de synthèse, mais considérant que chaque main digne d'être testée et entraînée est attestée dans de nombreuses pages, il a semblé bien plus économique en termes d'ingénierie d'éviter le doublonnement des fichiers plutôt que de travailler à la réconciliation des transcriptions³⁹.

1.4.2 Établir des normes de transcription

Il faut évoquer brièvement ici les principes généraux de la transcription des textes, les normes détaillées étant reportées en annexe⁴⁰.

Il est primordial pour l'établissement de ces principes de rappeler que la reconnaissance automatique des écritures procède caractère par caractère. Elle ne tient compte ni du contexte syntaxique ni du contexte sémantique. Il n'est donc pas possible d'apprendre à un algorithme de reconnaissance à appliquer un accent sur la lettre *a* lorsqu'il s'agit d'une préposition, ni de lui apprendre à reconnaître la lettre *é* avec accent aigu dans le mot *décoration*. Si l'accent a été omis par le scribe, transcrire *é* à la place de *e* consiste à apprendre à l'algorithme que, par ailleurs, le mot *vie* devrait être transcrit *vié*.

La démarche de reconnaissance automatique de l'écriture peut être envisagée de plusieurs manières, soit comme une imitation des caractères écrits de la sources (où l'on respecte les abréviations sans les développer et où l'on imite la forme des lettres⁴¹), soit comme une transcription déjà interprétative de la source qui uniformise les allographes

39. Le script Python d'examen des doublons était suffisamment bref pour être écrit nativement dans le notebook *Tester et entraîner un modèle de reconnaissance d'écriture* (S. Biay, *Tester et Entraîner Un Modèle de Reconnaissance d'écriture...*) ; on le trouve sous le titre *Classer les images par mains*

40. Cf. B, p. 51.

41. Cette méthode de transcription est généralement dénommée allographétique ; cf. Dominique Stutzmann, « Paléographie statistique pour décrire, identifier, dater... Normaliser pour coopérer et aller plus loin ? », dans *Kodikologie und Paläographie im digitalen Zeitalter 2 - Codicology and Palaeography in the Digital Age 2*, dir. Franz Fischer, Christiane Fritze et Georg Vogeler, Norderstedt, 2011, t. 3, p. 247-277, URL : <https://kups.ub.uni-koeln.de/4353/> (visité le 08/01/2022), p. 250 et *passim*

et restitue les abréviations⁴². En réalité, aucune de ces options ne peut être appliquée de façon radicale de bout en bout d'une transcription, chacune rencontrant des limites dans son applicabilité.

Par exemple, le projet Notre-Dame de Paris et son cloître (e-NDP) aborde l'entraînement des algorithmes de reconnaissance d'écriture dans l'optique de leur apprendre à restituer les abréviations des scribes des sources du chapitre⁴³. Cette démarche, qui permet de faire l'économie d'une phase de développement des abréviations, utile à l'interrogation du texte par un moteur de recherche, trouve néanmoins une limite dans la capacité des modèles de reconnaissance d'écriture à restituer plusieurs lettres pour un seul caractère abrégé⁴⁴.

A contrario, la volonté d'imiter au plus près les usages scribaux se heurte notamment aux difficultés des modèles à reconnaître les espaces. La tendance de certains scribes à coller certains mots les uns aux autres, ou plus encore à détacher quelque peu les parties d'un même mot entraîne l'omission ou la transcription d'espaces erronée du point de vue de la lecture interprétative du texte. Une stricte imitation des usages scribaux devrait conduire à respecter ces phénomènes lors de l'établissement des transcriptions de test et d'entraînement, et ce avec deux inconvénients de taille : la difficulté d'apprécier la réalité dimensionnelle d'un espace (à partir de quelle quantité de blanc une espace doit être transcrise) et le travail fastidieux mais indispensable de restituer ultérieurement le juste espacement du texte pour en permettre une restitution propre à la lecture ou à l'analyse.

L'indispensable compromis à trouver sur ce point a été guidé par les réflexions menées dans le cadre du séminaire *Création de modèle(s) HTR pour les documents médiévaux en ancien français et moyen français entre le X^e-XIV^e siècle*⁴⁵. On s'est donc efforcé de définir le degré d'imitation de la source manuscrite conforme aux besoins spécifiques de l'édition de la correspondance de C. de Salm, en suivant autant que possible les choix génératifs pronés par la communauté scientifique consituée autour du projet Cremma et qui correspondent de manière tendancielle au concept de transcription *graphématisque* traduit et expliqué par D. Stutzmann⁴⁶.

Ainsi, la restitution des allographes a été écartée dans la mesure où le seul exemple d'allographe contenu dans les documents du projet est le *s* long. D'autre part, la diffi-

42. C'est le cas de la transcription dite diplomatique ; cf. Olivier Guyotjeannin, Jacques Pycke et Benoît-Michel Tock, *Diplomatique médiévale*, 1993^e éd., Turnhout, 2006 (L'atelier du médiéviste, 2)

43. Sergio Torres Aguilar, « e-NDP (Notre-Dame de Paris et son cloître) : 26 registres du chapitre de Notre-Dame de Paris datés du 14e-15e en latin (principalement) et français », dans *Transkribus / eScriptorium : transcrire, annoter et éditer numériquement des documents d'archives, ateliers à la BnF*, Paris, BnF, site François-Mitterrand, 2022.

44. Le constat d'une incapacité des modèles à restituer plus de deux ou trois lettres a été formulé dans la discussion consécutive à la présentation citée dans la note précédente.

45. A. Pinche, Séminaire "Création de modèle(s) HTR pour les documents médiévaux en ancien français et moyen français entre le X^e-XIV^e siècle" : compte-rendu de la séance n° 3, CREMMALAB, 2021, URL : <https://cremmalab.hypotheses.org/compte-rendu-seance-3> (visité le 13/06/2022).

46. D. Stutzmann, « Paléographie statistique pour décrire, identifier, dater..... », p. 251.

culté et le coût impliqué par l'imitation de l'espacement des mots a été tranchée par la restitution de l'espacement moderne des mots dès lors que l'usage scribal ne caractérisait pas un usage établi ; en revanche, les élisions, agglutinations ou encore les lexicalisations (consacrées ou fautives) ont été respectées : *d'avantage*, *C'a été*, *tédeum*. L'usage scribal a également été respecté dans l'accentuation des caractères, l'abréviation des mots, l'usage des majuscules, l'orthographe et la ponctuation.

Ces choix se justifient d'une part en ce qu'ils permettent un bon entraînement de la reconnaissance d'écriture caractère par caractère, d'autre part en raison de leur correspondance avec les critères de l'édition finale du texte, qui respecte les usages scribaux jusque dans l'application non systématique des règles d'accentuation des mots.

Enfin, concernant les passages biffés, les palimpsestes ou encore les passages illisibles, un ensemble de solutions d'encodage a été proposé dans le cadre du projet DAHN⁴⁷. Plutôt que d'introduire des caractères tels que £, €, etc. dans les transcriptions, ce qui les éloignerait d'une transcription de type graphématisé et limiterait les possibilités de réutilisation éventuelles de ces transcriptions, on a préféré appliquer les conventions préconisées par la convention de Leyde⁴⁸, retenues dans le cadre du Cremma⁴⁹ (cf. p.52).

Lorsqu'une correction est inscrite dans l'interligne, la ligne interlinéaire reçoit une annotation spécifique qui permet de l'identifier comme telle au moment de l'encodage⁵⁰.

1.4.3 Transcription manuelle *versus* transcription automatique : quelle bonne méthode pour l'entraînement ?

Il a été question plus haut des méthodes progressive et récursive d'entraînement⁵¹.

La méthode récursive a été testée au cours du stage pour constituer des transcriptions d'entraînement ; on faisait appel de surcroît à la correction semi-automatisée des transcriptions dans le but de gagner du temps... pour finalement revenir à la transcription manuelle. La correction automatisée ne permettant pas d'obtenir un résultat, il fallait en réalité corriger deux fois la même page. Et même en se limitant à une correction purement manuelle des transcriptions automatiques, la nécessité de suivre les usages scribaux (notamment l'accentuation des mots) imposait un contrôle visuel plus intense pour corriger une transcription déjà faite que pour produire cette transcription.

47. F. Chiffolleau, *Few Tips for Reading the Text Files*, DAHN Project, URL : <https://github.com/FloChiff/DAHNProject/tree/master/Project%20development/Texts> (visité le 11/04/2022).

48. « Leiden Conventions », dans *Wikipedia*, 2021, URL : https://en.wikipedia.org/w/index.php?title=Leiden_Conventions&oldid=1004624327 (visité le 05/05/2022).

49. A. Pinche, *Séminaire "Création de modèle(s) HTR pour les documents médiévaux en ancien français et moyen français entre le Xe-XIV^e siècle" : compte-rendu de la séance n° 2*, CREMMALAB, 2021, URL : <https://cremmalab.hypotheses.org/seminaire-creation-de-modeles-htr/compte-rendu-de-la-seance-n-2> (visité le 05/05/2022).

50. Voir *infra* p. 27.

51. Voir *supra* 1.3.6, p. 27.

1.4.4 Éliminer d'une transcription les lignes attestant des écritures parasites

La présence de plusieurs écritures dans la même image est problématique pour évaluer la capacité d'un modèle à reconnaître une écriture particulière, car la présence d'une écriture différente dans la même page est de nature à parasiter cette évaluation. Or la phase de segmentation de la page, qui permet la reconnaissance de toutes les lignes d'écriture, ne peut pas être paramétrée pour ignorer une ou plusieurs écritures déterminées. Une fois toutes les lignes de l'image reconnue, il est donc nécessaire de supprimer les lignes que l'on juge parasites.

Si l'on veut procéder de façon manuelle en supprimant les lignes une par une dans l'interface eScriptorium, l'opération peut se révéler fastidieuse : supprimer une page entière consistera à cliquer sur un minimum de trente lignes... Un script a donc été développé pour faciliter ce travail⁵². La transcription manuelle que l'on effectue sur les seules lignes attestant l'écriture que l'on souhaite tester laisse toutes les autres lignes de la page vides. Le script transforme l'export de cette transcription (format ALTO) et supprime de celle-ci toutes les lignes laissées vides. On peut dès lors tester un modèle de reconnaissance d'écriture avec la certitude que celui-ci ne tentera pas de reconnaître une écriture dans des zones de l'image où on ne le souhaite pas.

1.4.5 Comparer les performances des modèles

On a procédé à la comparaison des performances de plusieurs modèles en utilisant le logiciel libre Kraken en ligne de commande⁵³ (les entraînements ont été également effectués à l'aide de ce logiciel).

Afin d'éviter une surévaluation des performances du modèle entraîné de zéro par H. Souvay⁵⁴, les performances de ce dernier ont été évaluées à partir de transcriptions nouvellement produites. L'acuité de reconnaissance de l'écriture sur l'unique main attestée dans le corpus d'entraînement de ce dernier a atteint 77,25% seulement.

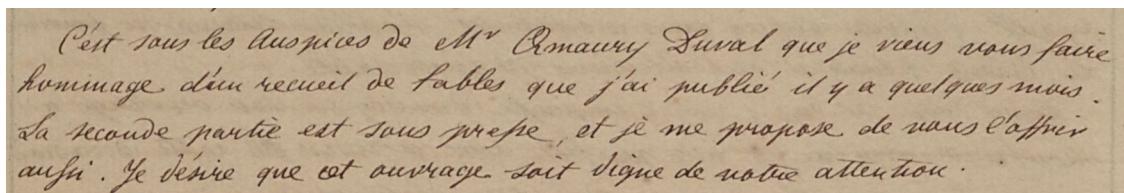


FIGURE 1.18 – Copie d'une lettre de Pierre-Augustin Rigaud à C. de Salm, le 13 avril 1824.

52. S. Biay, *supprLignesVides.Py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/supprLignesVides.py> (visité le 31/05/2022).

53. *Kraken [Documentation]*...

54. H. Souvay, *La Correspondance de Constance de Salm (1767-1845) : Rapport de Stage...*

Voici la prédiction correspondante :

cet ju le gusaires de Mr touauz ctral ne ; qu venes tat
 bonmage Liu rececit de fables que j'li puslié il y a quelsurs mos
 L ronde partie est sous presse. et je me prapose de vous l'affrir
 aussi. Je dhrre qe cet ouvrage sait digne de votre atenson⁵⁵

Cette acuité était supérieure à celle atteinte par le modèle générique du projet Lectaurep, entraîné sur des écritures administratives du XIX^e siècle (73,12%), mais elle était en revanche inférieure à celle atteinte par le modèle affiné sur les contrats de mariage à partir d'un premier modèle mixte dans le cadre du même projet, qui atteignait quant à lui une acuité de 80,42%⁵⁶ :

Cest saus les Auopices de M^r Amaury Duval rue je vieus mous favre hommage
 dem recueit de lables que ai publie et & a quetques mais la seconde partie est
 sons prepe, et se me propose de nans l'apnis aufri. Je désire que et auvrage soit
 sique de nobre attention -

Battu sur sa propre écriture d'entraînement, le modèle entraîné de zéro par H. Souvay a donc été immédiatement délaissé pour privilégier le modèle Lectaurep affiné sur les contrats de mariage, dont l'acuité s'est révélée meilleure sur toutes les mains que l'on a eu l'occasion de tester. Comme on peut le constater à l'œil nu, une acuité de 80% reste très insuffisante pour rendre le texte exploitable. Mais il était évident que la meilleure progression serait obtenue en entraînant ce même modèle à reconnaître une variété d'écriture des scribes de la correspondance de C. de Salm.

Déterminer le nombre de pages transcrrites nécessaires à l'entraînement efficace d'un modèle de reconnaissances des caractères dépend d'une multiplicité de facteurs, au premier rang desquels on trouve la régularité de l'écriture et son degré de cursivité. Entrent également en ligne de compte les performances de modèles déjà produit sur des écritures similaires (mais combien proches ? c'est toute la question), la densité d'usage des abréviations ou encore la qualité des reproductions photographiques. La qualité de l'encre et le contraste entre l'encre et la page sont également de nature à influencer la taille du corpus d'entraînement nécessaire pour parvenir à de bonnes performances⁵⁷.

En procédant à la constitution d'une vérité de terrain d'une dizaine de pages⁵⁸ pour

55. Notice d'inventaire : *CdS/02_3/056*, URL : <https://constance-de-salm.de/archiv/#/document/8885>.

56. Les modèles hérités de ce projet sont disponibles sur un dépôt ouvert : *Kraken Models : Transcription Models*, GitLab Inria, URL : <https://gitlab.inria.fr/dh-projects/kraken-models/-/tree/master/transcription%20models> (visité le 28/04/2022). Les versions utilisées sont : *generic_lectaurep_26* et *cm_ft_mrs15_11*.

57. P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The eScriptorium VRE for Manuscript Cultures »...

58. Cette mesure est elle-même à relativiser car la densité de l'écriture sur nos documents est variable et certaines pages ne sont pas complètement remplies ; il a donc fallu compenser les blancs par des pages supplémentaires, et au final cette mesure d'une dizaine de pages est des plus approximatives.

chacune des mains sélectionnées, des scores supérieurs à 95% d'acuité ont été atteints dès le premier entraînement :

- 1re main de la seconde copie des lettres⁵⁹ : 98,68%
- 3e main de la seconde copie des lettres⁶⁰ : 96,31%
- 1re main de la correspondance Martini⁶¹ : 97,88%
- 2e main de la correspondance Martini⁶² : 96,27%

Voici la prédiction, où les quelques fautes rémanentes ont été colorées en rouge :

C'est sous les Auspices de Mr Amaury Duval que je viens vous faire
hommage d'em recueil de lables que j'ai publié il y a quelques mois
La seconde partie est sous presse, et je me propose de vaus l'assrir
aussi. Je désire que cet ouvrage soit digne de votre attention.

La 2e main de la seconde copie des lettres (mainCdS02_Konv002_02) a été écartée des entraînements afin de constituer un témoin complémentaire des performances du modèle. On a ainsi pu constater le gain de souplesse du modèle entraîné, c'est-à-dire l'amélioration de sa capacité à reconnaître des écritures pour lesquelles il n'a pas été entraîné. L'acuité sur cette main a progressé de 73,09% avant l'entraînement sur les quatre autres mains à 91,54% après cet entraînement.

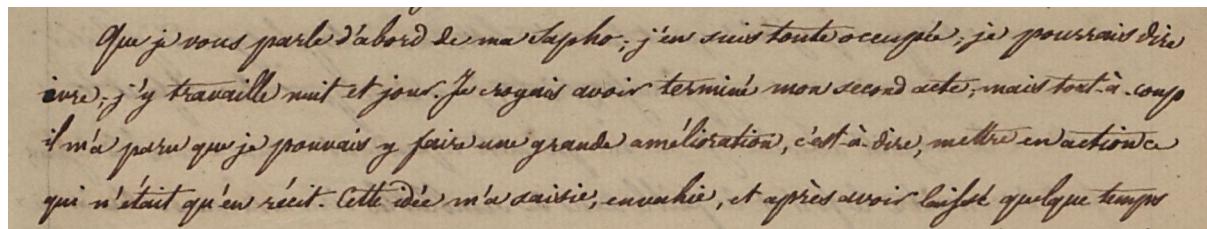


FIGURE 1.19 – Copie d'une lettre adressée par C. de Salm à Jean-François Thurot, le 21 février 1794.

Que j vous parle d'abord le ma Saphe ; j'en suis toute occuple je pourrais drie
êvre j'y travaille nuit et jour. Jecrogais avoir terminé mon second acte, mais toet à-coup
lma paru que je pouvais y faire une grande amélisration, c'est à Pire, mettre en actionce
qui n'était qu'en récit. Cette dée m'a saisie, envuhie, et après devoir laifst quelque tempps⁶³

Comme on peut le constater avec cette prédiction, une acuité de 91% laisse encore une lourde tâche de correction à l'éditeur du texte pour parvenir à un résultat publiable. Même si ce pourcentage peut sembler élevé, il reste indispensable de procéder à l'entraînement du modèle de reconnaissance pour chaque nouvelle main afin de dépasser le score de 95% au-delà duquel la correction de la graphie des mots devient légère (la ponctuation et l'accentuation restant à examiner de près).

59. Dénommée mainCdS02_Konv002_01.

60. Dénommée mainCdS02_Konv002_03.

61. Dénommée mainCdS02_Konv019_01.

62. Dénommée mainCdS02_Konv019_02.

63. Notice d'inventaire : CdS/02_1/031-032, URL : <https://constance-de-salm.de/archiv/#/document/8440> (visité le 13/06/2022).

1.4.6 Tenir un journal des résultats de tests et d'entraînements

Les performances du nouveau modèle, dénommé `cds_lectcm_04_mains_01`⁶⁴, n'avaient pas été espérées aussi bonnes. La démarche de tenue d'un journal de test et d'entraînement avait donc été développée en prévision de la nécessité de répéter les entraînements et de suivre la progression des performances. Dans cette optique, un script Python a été écrit pour pré-remplir un journal de résultats⁶⁵.

Ce script analyse le contenu des dossiers de test et d'entraînement pour lesquels des préconisation de nommage et d'organisation ont été formulées plus haut ; il enregistre la date et l'heure du moment, dénombre les dossiers de mains et récupère leurs labels, il dénombre également le nombre de fichiers contenus dans les vérités de terrain (dossiers *train*) de chaque main et permet ainsi de suivre l'accroissement de tel ou tel sous-corpus d'entraînement au fil des opérations. Les résultats de tests du nouveau modèle sur les différentes mains doivent ensuite être inscrits manuellement dans le fichier.

Ce script permet également de conserver une trace de la distribution des fichiers où les mains sont attestées et de la liste de ceux qui composent les corpus de test et d'entraînement. Ces listes constituent ainsi l'archive détaillée des tests et des entraînements que l'on a effectués. Elles permettent la suppression de l'arborescence du dossier d'entraînement que l'on a élaboré sans perte d'information et garantissent la transparence des données d'entraînement du modèle⁶⁶.

Comment procéder à de nouveaux entraînements pour adapter le modèle de reconnaissance à d'autres mains de la correspondance ? En théorie, il serait plus indiqué de poursuivre l'entraînement de modèle par l'enrichissement du corpus déjà constitué et la réitération des entraînements que l'on a effectués. Recommencer en somme les entraînements que l'on a effectué à partir d'un corpus plus riche. Cette option est celle qui garantit la plus grande générnicité de modèle. Mais une autre méthode peut naturellement être envisagée : repartir du modèle que l'on a produit et l'affiner avec des données nouvelles. Cette méthode peut nuire quelque peu à la générnicité du futur modèle (bien que nous n'ayons pas eu la possibilité de tester ce point) mais elle permet de réduire considérablement le temps de calcul des entraînements. Ce processus extrêmement gourmand en temps de calcul (et très dépendant des performances de l'ordinateur utilisé), sera sérieusement écourté si l'on se contente d'un affinage par quelques données supplémentaires.

64. Cette dénomination signifie : C. de Salm ; Lectaurep, contrats de mariage ; quatre mains ; version 1.

65. S. Biay, *journalReconn.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/journalReconn.py> (visité le 25/05/2022).

66. Le fichier contenant ces données se trouve à l'adresse suivante : <https://github.com/sbiay/CdS-edition/blob/main/htr/mains/mains.json>.

1.4.7 Injecter les transcriptions manuelles dans les prédictions

Le test et l'entraînement des modèles de reconnaissance d'écriture impose la production de transcriptions manuelles du texte. Il nous est apparu essentiel que cette tâche un peu fastidieuse soit pleinement valorisée dans le processus d'édition et que ces transcriptions théoriquement parfaites servent non seulement à l'entraînement des modèles mais soient aussi exploitées pour la production de l'édition finale.

La méthode la plus simple pour joindre les fichiers ALTO contenant les transcriptions manuelles aux fichiers contenant la prédiction automatique du texte des autres pages d'un même dossier est de regrouper ces fichiers ensemble. Or, nous avons voulu tenir compte de la possibilité que les transcriptions manuelles ne recouvrent pas toutes les lignes d'écriture d'une page – le cas n'est pas très fréquent, mais nous y avons été confronté. Certaines mains n'étant attestées que de manière sporadique, en compagnie d'autres écritures, la méthodologie d'entraînement impose de ne transcrire que l'écriture propre au test ou à l'entraînement, laissant les écritures voisines de côté. Il résulte de cette nécessité que les fichiers ALTO contenant les transcriptions manuelles peuvent être lacunaires : ils ne peuvent donc pas se substituer aux fichiers contenant la prédiction complète des lignes d'écriture d'une page au risque de remplacer une partie des prédictions par du vide.

Il était donc nécessaire de concevoir une méthode de remplacement, dans les fichiers contenant la prédiction automatique du texte, des seules lignes pour lesquelles nous avions produit des transcriptions manuelles. Cibler de manière précise des lignes d'écriture dans un fichier ALTO est rendu possible par l'identifiant unique de chaque élément contenant une ligne de texte (`TextLine`). Nous avons donc écrit un script python⁶⁷ capable d'analyser toutes les lignes d'écriture des fichiers de nos vérités de terrain et de comparer leur identifiant avec ceux des lignes des fichiers des prédictions automatiques portant les mêmes noms. En cas de correspondance entre les identifiants, la transcription manuelle vient remplacer la prédiction du texte.

1.5 La correction semi-automatisée

Une fois que l'on dispose d'un modèle de reconnaissance d'écriture suffisamment bien entraîné pour donner des prédictions satisfaisantes pour toutes les mains principales d'une source, on peut réaliser des prédictions sur l'ensemble de la source.

Même avec un modèle très performant, le travail de correction des fautes rémanentes ne peut être négligé. Son automatisation permet de gagner un peu de temps ; elle joue surtout le rôle de tamis, attirant l'attention de l'éditeur sur les graphies inhabituelles des mots là où son œil pourrait les laisser échapper.

⁶⁷. Id., *injectTranscript.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/injectTranscript.py> (visité le 03/06/2022).

Mais automatiser la correction des prédictions requiert de la prudence. Il faut naturellement veiller à ne pas remplacer involontairement des prédictions justes, et comme les règles d'édition que l'on applique suivent de près les usages scribaux, la graphie des mots ne saurait être uniformisée. La notion de justesse doit donc être élargie aux variations graphiques de chaque scribe. De ce point de vue, l'automatisation des corrections peut s'avérer précieuse pour signaler à l'éditeur un usage scribal particulier, comme l'omission d'un accent aigu sur le mot *redaction*, un point dont le contrôle est nécessaire bien qu'il puisse très facilement échapper à l'attention.

L'automatisation des corrections ne consiste donc pas à remplacer automatiquement le contenu des prédictions mais à analyser ce contenu et à signaler les mots représentant un problème, et si possible à proposer une correction que l'éditeur sera libre d'appliquer ou non.

Le résultat de cette opération est imparfait ses limites sont discutées ci-après. On attend d'elle qu'elle accompagne et facilite la correction de la prédition par l'éditeur, mais pas qu'elle produise un texte ayant le statut de vérité de terrain ou de texte établi. Par conséquent, cette correction n'intervient pas dans le processus d'entraînement d'un modèle HTR. Une fois les modèles HTR correctement entraînés, elle permet de résoudre un certain nombre d'erreurs en amont de la transformation des prédictions au format ALTO vers le format TEI, où une correction manuelle approfondie du texte est nécessaire pour son établissement définitif.

Nous avons suivi la démarche explicitée dans la documentation du projet DAHN⁶⁸ et proposé quelques développements aux scripts issus de ce projet⁶⁹.

1.5.1 Trouver le bon compromis entre granularité et performance

Les qualités respectives de plusieurs méthodes ont été évaluées afin de d'établir les paramètres les plus intéressants pour cette phase du travail. On a évoqué précédemment l'impossibilité d'un résultat parfait. L'automatisation des corrections ne permet absolument pas de faire l'économie d'une révision approfondie du texte par l'éditeur. Elle doit par conséquent faire preuve d'un haut degré de performance : son but est d'abord et avant tout de faire gagner du temps. Or, chaque forme signalée au cours de cette phase requiert une décision de l'éditeur : ces formes doivent donc être très pertinentes afin de ne pas

68. F. Chiffolleau, *How to Do a Post-OCR Correction for TEXT Files*, DAHN Project, 8 avr. 2022, URL : <https://github.com/FloChiff/DAHNProject/blob/master/Project%20development/Documentation/Post-OCR%20correction%20for%20TEXT%20files.md> (visité le 11/04/2022).

69. Le script principal porte le nom de *spellcheckTexts* : S. Biay et F. Chiffolleau, *spellcheckTexts.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/spellcheckTexts.py> (visité le 19/04/2022). Ce script est fondé sur l'utilisation du module publié par Tyler Barrus, *Pyspellchecker : Pure Python Spell Checker Based on Work by Peter Norvig*, version 0.6.3, URL : <https://github.com/barrust/pyspellchecker> (visité le 19/04/2022).

gaspiller le temps de ce dernier. Contrôler chaque mot dans son contexte serait largement contre productif.

Il est donc très vite apparu nécessaire de ne pas signaler à l'éditeur les mots dont la graphie a été validée par ailleurs. Pour cela, on s'est appuyé d'une part sur un dictionnaire généraliste de la langue française et d'autre part sur les mots de la correspondance-même de C. de Salm, à savoir les mots contenus dans les vérités de terrain que l'on a produites pour le test et l'entraînement des modèles de reconnaissance d'écriture⁷⁰.

En résumé, la correction automatisée se concentre sur l'orthographe des mots. Elle ne traite pas la ponctuation. De plus, elle considère qu'une forme présente dans les vérités de terrain ou dans le dictionnaire de référence de la langue française est en soi valide. Ainsi, elle ne signale pas les mots mal prédits dont l'orthographe est attestée ailleurs dans les vérités de terrain ; par exemple, dans la prédiction *Dans vu siècle où tous les talens...*, la prédictiton erronée *vu* pour *un* ne sera pas signalée, car le mot *vu* est attesté ailleurs.

1.5.2 Analyser les mots

Le script procède à une recherche de correspondances entre les formes du texte et un dictionnaire de référence par des permutations de lettres : il est en mesure de proposer des formes considérées comme justes dans une limite de deux fautes par mot. Par exemple, il reconnaît que la meilleure proposition pour le mot *deusx* est *deux*, mais n'est pas capable d'associer la forme *pubiès* aux mots de la famille de *publier*.

Afin de faciliter la correction des dictionnaires générés par le script pour chaque page (ce sont ces dictionnaires qui permettent de valider les propositions de correction), on a développé le script pour afficher le contexte du mot et en conserver la mémoire, ce qui limite le besoin d'allers-retours entre le dictionnaire à corriger et l'image ou la prédiction d'origine.

Une fois les corrections validées, un second script écrit par F. Chiffoleau permet de les appliquer aux fichiers contenant les textes⁷¹. Originellement conçu pour remplacer des chaînes de caractère n'importe où dans le fichier concerné, il faisait courir le risque de remplacements abusifs. Par exemple, si la forme *natur* devait être corrigée en *nature* et que la même page de texte contenait aussi le mot *naturellement*, une application globale des corrections entraînerait la création d'une faute : *naturellement* deviendrait *natureellement*. Le script a donc été perfectionné afin de procéder à l'application des corrections ligne par ligne et mot par mot⁷².

70. Pour exploiter ce second réservoir de mots, une fonction appelée *collecteMots* a été ajoutée au script principal.

71. S. Biay et F. Chiffoleau, *textCorrection.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/textCorrection.py> (visité le 19/04/2022).

72. Pour l'application mot par mot, on a utilisé le module *SpaCy : Industrial-strength Natural Language Processing in Python*, URL : <https://spacy.io/> (visité le 27/04/2022).

En outre, il s'est avéré nécessaire de modifier la méthode d'application des corrections aux fichiers ALTO des prédictions en optant pour l'écriture d'un authentique arbre XML et non d'une imitation d'arbre au format `txt`, comme c'était le cas dans le script d'origine⁷³.

1.5.3 Gérer les résolutions ambiguës

Appliquer des scripts de correction automatique, on l'a signalé plus haut, comporte le risque d'appliquer partout des corrections ne se justifiant que dans certains cas et ainsi de générer des fautes. Le problème de l'ambiguïté des corrections se pose lorsqu'une prédition peut se prêter selon le contexte à plusieurs résolutions différentes : par exemple la forme *cele* peut résulter tantôt de l'oubli d'un *l* (on corrigera en *celle*), tantôt de la reconnaissance d'un *e* à la place d'un *a* (on corrigera en *cela*).

Dans un premier temps nous avons procédé selon une méthode d'automatisation qui neutralisait les corrections ambiguës : *cele* était intégré à la liste globale des corrections avec une absence de lemme afin d'être exclu de la correction automatique.

Cette méthode présentait plusieurs inconvénients :

- Une fois que l'on avait procédé à des corrections pour les mots d'une page, le script qui les intégrait au fichier rassemblant toutes les corrections contrôlait qu'une forme ne puisse pas être associée à plusieurs corrections. Lorsqu'une ambiguïté était repérée, il fallait intervenir sur les deux fichiers pour neutraliser la correction. Devenu fréquent, ce processus diminuait le bénéfice de temps attendu de la correction automatique ;
- D'autre part, il s'est avéré que les corrections ambiguës sont nombreuses, car il suffit d'une faute sur un petit mot pour le rendre ambigu avec un autre mot : *ue* peut être corrigé en *rue* ou en *une*; *veu*s peut être corrigé en *veux* ou en *vou*s; *ceste* peut être corrigé en *cesse* ou en *cette*.

Plutôt que neutraliser la correction de ces mots, il s'est donc avéré nécessaire de prendre en charge ces ambiguïtés. Mais se contenter de lister des propositions de correction de manière indiscriminée aurait pu là encore nuire aux performances de l'opération. Afin de faciliter la sélection de la bonne correction parmi une liste de propositions, une nouvelle fonction a été écrite⁷⁴ dont le rôle est de classer les mots attestés dans les vérités de terrain par ordre décroissant de nombre d'occurrences. Ainsi, le mot le plus fréquent est toujours proposé comme premier choix au correcteur, ce qui maximise les chances qu'il n'ait pas à

73. L'injection des transcriptions manuelles en lieu et place des prédictions (cf. ci-dessus, 1.4.7, p. 39) dans les seuls fichiers appartenant au corpus d'entraînement de la reconnaissance d'écriture a entraîné une modification irrémédiable de l'indentation de ceux-ci. L'indentation de ces fichiers étant devenue différente des autres fichiers des prédictions, il n'était plus possible de s'appuyer sur l'identité des indentations pour repérer les lignes de textes à remplacer. Il devenait donc obligatoire de s'appuyer sur la hiérarchie de l'arbre XML pour appliquer ces corrections.

74. Il s'agit de la fonction dénommée *ordreOccurrences*; cf. Id., *spellcheckTexts.py*...

intervenir sur la correction à effectuer.

1.5.4 Élaborer et enrichir un nouveau dictionnaire de la langue française

La capacité de l'application Python Pyspellchecker à analyser les formes des mots repose sur des dictionnaires numériques spécifiques à chaque langue. Comme l'indique la documentation de l'application⁷⁵, ces dictionnaires ont été élaborés à partir de la collecte massive de formes de mots parmi les ressources du site OpenSubtitles⁷⁶, qui fournit des fichiers de sous-titres pour les œuvres cinématographiques dans de très nombreuses langues.

La récolte lexicale qui découle de cette source est extrêmement vaste. Pour la langue française, le nombre de formes collectées est de presque 800 000 ! Il est important de rappeler qu'il s'agit de formes et non de lemmes : on y trouvera pour le verbe *aimer* : aime, aimes, aimer, aimons, aimez, aiment, etc.

Il n'a pas été possible de découvrir comment OpenSubtitles rassemble ses sources textuelles, mais il est assez évident que la principale origine de ces sous-titres sont les fichiers contenus dans les supports DVD et Blu-Ray. Certains de ces fichiers sont en outre issus de la traduction automatique des sous-titres d'une langue source vers une langue cible, ce dont résultent potentiellement des données de piètres qualité.

Il n'est guère possible d'analyser ici de manière précise la nature de ces données, mais les hypothèses que l'on vient de formuler sont tout à fait susceptibles d'expliquer la médiocre qualité des formes lexicales croisées dans le dictionnaire du français utilisé par Pyspellchecker. Si l'on se tourne vers les formes les plus rarement dénombrées dans ce dictionnaire (celles comptant 1 ou 2 occurrences dans tout le corpus représentent plus de la moitié du dictionnaire), force est de constater la piètre qualité des données. Voici un tout petit extrait des premières formes lexicales comptabilisant deux occurrences :

phiiiy, tetsujiro, étreins-le, rugissons, causatif, armonia, qccupe-toi, découvez, masanté, jannelke, aleksi, qpidon, 500km, unejeunesse, birnholz-vazquez, traînons-nous, peterkins, koidry, vinitt, mentait., bonne-journée, micromonde, myélogène, uilise, 313e, rubindium, ddeokbeoki, 'irlandia, donie, brichelle

L'inconvénient pour l'analyse des prédictions automatiques de la correspondance de C. de Salm est double. La quantité énorme de formes sémantiques coûte du temps à la recherche automatique des fautes. Mais plus grave, les innombrables bizarries que l'on trouve dans ce dictionnaire finissent nécessairement par parasiter l'analyse de nos prédictions. Nous avons ainsi fait l'expérience que la forme *ette*, qui n'existe pas en français et pourtant est attestée 36 fois dans le dictionnaire francophone de Pyspellchecker (!) a

75. T. Barrus, *Pyspellchecker*...

76. URL : <http://www.opensubtitles.org/>

de fait été considérée comme juste, passant à travers les mailles du filet de la correction automatique.

Annexes

Annexe A

Transcriptions de deux manuscrits autographes de C. de Salm

A.1 Premier extrait

La ponctuation a été quelque peu modernisée pour rejoindre une édition de type diplomatique.

Extrait du début de la lettre de C. de Salm à Therese Thurn und Taxis du 20 mai 1825¹ :

Dyck, ce 20 mai 1825.

Madame,

Que vous dire de mon silence ?
Comment pourrai-je l'expliquer ?
je n'en sais rien : le travail, la souffrance
le repos ; est ennui qui vient tout attaquer,
fruit de longues douleurs, dont la première me
semble ††††ante pour jamais
le charme d'une douce et simple jouissance,
voilà pourquoi, si j'en crois l'apparence,
depuis si longtemps me tais.
Cependant, je dois vous le dire,
moi même je ne puis bien décider ce point ;
car si je ne vous écris point,
à chaque instant, je voudrais vous écrire.

1. CdS/67/022-030, URL : <https://constance-de-salm.de/archiv/#/document/3814> (visité le 13/06/2022).

Mais le Printems, son éclat, sa fraicheur,
La nature si belle en ses jours des pleud†††
par leur vivifiante flamme
de mon Corps épuisé raniment les ressorts.
Ces jeunes fils, vrai soutiens de mon âme
Sans le savoir secondant ses efforts
de l'existance, aussi, me rouvrent les trésors
et charmeur de nouveau narcoi†
par luy de grands chaos d'esperances remplis.
Enfin le sort et plus juste et plus doux
pour un moment au moins de mes maux me soulage
je sens renaitre en moi le calme, le courage
je me retrouve et je reviens à vous.

Voici, Madame le tableau fidele de tout ce qui passe en moi depuis que je ne vous ai ecrit, et de tout ce que j'éprouve aujourd'hui. Mon ††††† en est fort triste. Ce n'est pas mon absence de Paris qui en est cause : Mon âge ; mes habitudes de travail ne me permettent pas de†††† cet' privation si vivement, C'est cet' vieille douleur qui est toujours ici, et aussi la perte d'une foule de mes amis et de personnes, de connaissance. Encore tout recemment j'ai vu disparaître Derrou, la P(rin)cesse Borghese avec qui j'avais été tres liée, et qui etait une aimable personne, et le malchanceux courrier, assassiné près de son chateau, dieu sait par qui ! (vous auriez vu ce malheur dans les journaux). Ce que l'on dit sur les causes de ce terrible èvènement est affreux † pa††r, et je n'ose l'écrire. [...]

A.2 Second extrait

Extrait du début de la lettre de C. de Salm à Fürst von Hatzfeldt du 2 mars 1828² :

Dyck ce 2 Mars 1818.

Vous serez sans doute surpris, Prince, de recevoir une lettre de moi dans ce moment, et je suis surprise aussi, d'avoir a vous l'ecrire sur le sujet dont je vais vous entretenir ; mais ayant tant de fois pris la plume pour des choses qui m'étaient étrangères, je ne vois pas pourquoi je ne la prendrais pas dans une occasion qui m'intéresse si personnellement, surtout quand je m'adresse à quelqu'un dont les sentiments de justice et d'amitié me sont également connus. Voici le fait : un de mes amis ayant appris, par hazard que Mme. Valentine avait le projet de troubler vot' tranquilité, s'est hâté de m'en prévenir, en me donnant à ce sujet des détails auxquels je l'avoue j'ai eu peine à croire. je n'attachais même à cet écrit aucune importance réelle ; mais mon mari n'a pu se refuser à me laisser lire, dans ce es ##### vos lettres, et celles de Mme. Valentine, et Comme j'ai vu dans vot' dernière que vous étiez mal informé sur les points les plus essentiels de ma position, j'ai cru sentir la nécessité de vous éclaircir moi-même, et de ne vous laisser rien ignorer de ce qui peut gêner vos idées sur moi. il n'est pas de rapport, Mons(seigneur), sous lesquels il ne me soit agréable d'avoir vot' estime entière, et celui dont il s'agit est sans doute, par une faveur qui se respecte, le plus essentiel de tous.

Il n'est ni dans mon caractère, ni dans ma manière d'agir d'attraper du malheur de qui que soit au monde ; je me suis fait, de tous tems, une loi de rester étrangère aux difficultés qui se sont élevées sans cesse, ent're Mme. valentine et mon mari, non quand j'ai pu [.....] l'obliger près de lui, ce dont je le prend à témoin. Quoi que les lettres asséz fréquentes qu'elle croit devoir lui adresser ne puissent m'être bien agréables, je me serais reproché d'y mettre le moindre obstacle et (soit-dit en passant), j'ai été blessée de la précaution qu'elle a prise [.....] de lui en faire remettre une par une voie détournée. Sure du cœur de mon mari, de mon état, de ma position, il ne m'est pas arrivé une seule fois de craindre l'effet de ces lettres, et j'ai poussé ce genre de procédés jusqu'à lui en envoyer une à Berlin, dans laquelle elle lui donnait un rendez-vous aux faux : mais je dois sortir de vot' indifférence lorsque je vois Prince, Mme. Valéline vous abuser, ou s'abuser au point de vous laisser croire que son divorce avec mon mari n'a pas été judicieux. [...]

2. C11/S92/047-049, URL : <https://constance-de-salm.de/archiv/#/document/765> (visité le 13/06/2022).

50 ANNEXE A. TRANSCRIPTIONS DE DEUX MANUSCRITS AUTOGRAPHES DE C. DE SALM

Annexe B

Normes de transcription

B.1 Accentuation

L'usage scribal a été respecté sans normalisation : en cas d'oubli de l'accent sur la préposition *à* on a transcrit *a*.

B.2 Majuscules et minuscules

La casse a été respectée sans appliquer les règles modernes : *je lis les Journaux Allemands*. Les accents ont été appliqués sur les majuscules.

B.3 Séparation des mots

La séparation des mots respecte l'usage graphique du scribe, mais sans imiter l'espace réel des mots. Ainsi, les élisions, agglutinations ou encore les lexicalisations (consacrées ou fautives) ont été respectées : *d'avantage*, *C'a été*, *tédeum*. Lorsqu'il n'y a aucun doute sur le fait que deux mots sont distincts, même s'il sont très proches dans l'espace de la page, ils ont été séparés d'une espace.

Nous n'avons pas restitué de trait d'union lorsque l'usage moderne l'imposerait : *portez vous bien*.

Dans le cas particulier de l'écriture personnelle de C. de Salm, les mots sont très souvent écrits dans un même mouvement de la plume. Dans ce cas seulement, ils ont été transcrits sans espace séparatrice.

B.4 Orthographe

L'orthographe des mots a été respectée : *enfans*, *momens*, *sentimens*, *cahos*.

Lorsque l'orthographe était erronée et changait la prononciation du mot, on a transcrit le mot sans le corriger : *Mr. Pron*s pour *Mr. Prou*s.

B.5 Abréviations

Les abréviations ont été transcrisées sans être résolues : *9bre* pour novembre, *Mr.* pour Monsieur.

L'abréviation *ll* pour livres (unité monétaire) a été transcrit par le caractère Unicode U + 1EFB.

B.6 Ponctuation

Les signes de ponctuation ont été transcrits fidèlement, y compris les points marquant une pause de la plume sans articulation syntaxique : *je ne sais pas . si vous en serez bien aise*. Les tirets ont été transcrits par le caractère *.*

B.7 Passages biffés, palimpsestes

Pour la transcription des phénomènes complexes tels que les passages biffés ou les palimpsestes, on a appliqué les conventions préconisées par la convention de Leyde¹, retenues dans le cadre du Cremma².

On a transcrit tout ce qui était lisible, y compris les lettres biffées, lorsque c'était possible, privilégiant le dernier état du texte et en plaçant le passage corrigé entre crochets : [abc].

On a remplacé chaque lettre biffée illisible par un point et placé l'ensemble des lettres concernées entre crochets : [...] (*pour deux lettres illisibles*).

B.8 Passages illisibles

Pour les problèmes de déchiffrement du texte, la convention de Leyde n'a pas d'autre préconisation que la mention en apparat³. Le choix a été fait de substituer à chaque lettre d'un mot non lu le signe †.

1. « Leiden Conventions »...

2. A. Pinche, *Création de modèles HTR : séance n° 2...*

3. *No sigla were suggested for corruptions (i.e. letters that are legible or restorable, but not understood). Instead, it was proposed that these should be dealt with in an apparatus* (« Leiden Conventions »...).

Glossaire

segmentation Analyse optique d'une image permettant d'obtenir la reconnaissance des régions et des lignes d'écriture.. 1, 5–7, 27, 28, 35

Acronymes

ALTO *Analyzed Layout and Text Object.* 6, 22, 35, 39, 40, 42

BnF Bibliothèque nationale de France. 6, 14

C. de Salm Constance de Salm. 3, 7, 8, 10, 11, 16, 29, 33, 36, 38, 41, 43, 51

Cremma Consortium Reconnaissance d'Écriture Manuscrite des Matériaux Anciens. 16, 33, 34, 52

DAHN Digital Edition of historical manuscripts. 3

DHIP Deutsches Historisches Institut Paris. 3, 16, 29

FuD Die Virtuelle Forschungsumgebung für die Geistes- und Sozialwissenschaften. 3

HTR *Handwritten Text Recognition.* 5, 16, 40

LAI Lettre Absente de l'Inventaire en ligne. 9, 10, 17, 20, 21

Lectaurep Lecture Automatique de Répertoires. 7, 8, 29, 36, 38

OCR *Optical Character Recognition.* 5, 16

SegmOnto SegmOnto : A Controlled Vocabulary to Describe the Layout of Pages. 19, 25, 26

TEI Text Encoding Initiative. 3, 18, 26, 40

XML Extensible Markup Language. 3

Bibliographie

Correspondance de C. de Salm

Cette liste contient les cotes de l'inventaire numérique de la correspondance, *Die Korrespondenz Der Constance de Salm (1767-1845). Inventar Des Fonds Salm Der Société Des Amis Du Vieux Toulon et de Sa Région Und Des Bestands Constance de Salm Im Archiv Schloss Dyck (Mitgliedsarchiv Der Vereinigten Adelsarchive Im Rheinland e.V.). Elektronische Edition*, 1^{er} avr. 2022, URL : <https://constance-de-salm.de> (visité le 11/04/2022) :

C11/S92/047-049, URL : <https://constance-de-salm.de/archiv/#/document/765> (visité le 13/06/2022).

CdS/02_1/001-330 : Correspondance générale, seconde copie, 1^{er} volume, 1785-1814, URL : <https://constance-de-salm.de/archiv/#/document/11215> (visité le 11/04/2022).

CdS/02_1/031-032, URL : <https://constance-de-salm.de/archiv/#/document/8440> (visité le 13/06/2022).

CdS/02_2/001-369 : Correspondance générale, seconde copie, 2^e volume, 1815-1821, URL : <https://constance-de-salm.de/archiv/#/document/11216> (visité le 11/04/2022).

CdS/02_2/073, URL : <https://constance-de-salm.de/archiv/#/document/8855>.

CdS/02_3/001-334 : Correspondance générale, seconde copie, 3^e volume, 1822-1828, URL : <https://constance-de-salm.de/archiv/#/document/11217> (visité le 12/04/2022).

CdS/02_3/056, URL : <https://constance-de-salm.de/archiv/#/document/8885>.

CdS/02_3/057-058, URL : <https://constance-de-salm.de/archiv/#/document/8887>.

CdS/02_3/058-059, URL : <https://constance-de-salm.de/archiv/#/document/8888>.

CdS/02_3/070-071, URL : <https://constance-de-salm.de/archiv/#/document/8907>.

CdS/19/036-037, URL : <https://constance-de-salm.de/archiv/#/document/10504>.

CdS/19/054-056, URL : <https://constance-de-salm.de/archiv/> (visité le 21/06/2022).

CdS/67/022-030, URL : <https://constance-de-salm.de/archiv/#/document/3814> (visité le 13/06/2022).

Valorisation du projet

BIAY (Sébastien) et SPYCHALA (Pauline), « L'intelligence Artificielle à l'IHA », dans *Hausinfo*, Paris, IHA, 2022.

Ressources du projet

- BIAY (Sébastien), *donneesImages.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesImages.py> (visité le 19/04/2022).
- *donneesNonPubliees.Py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesNonPubliees.py> (visité le 21/06/2022).
 - *injectTranscript.Py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/injectTranscript.py> (visité le 03/06/2022).
 - *journalReconn.Py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/journalReconn.py> (visité le 25/05/2022).
 - *Mains*, Éditer la correspondance de Constance de Salm (1767-1845), 10 juin 2022, URL : <https://github.com/sbiay/CdS-edition/tree/main/htr/mains> (visité le 10/06/2022).
 - *Préparer Le Traitement d'un Dossier*, 20 mai 2022, URL : https://github.com/sbiay/CdS-edition/blob/6c4e4d4cff3101a154b9fa7e4a248e7ac87ff7ee/htr/Preparer_le_traitement_dune_source.ipynb (visité le 23/05/2022).
 - *supprLignesVides.Py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/supprLignesVides.py> (visité le 31/05/2022).
 - *Tester et Entrainer Un Modèle de Reconnaissance d'écriture*, 20 mai 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/htr/Tester_et_entrainer_un_modele_HTR_avec_Kraken.ipynb (visité le 10/06/2022).
- BIAY (Sébastien) et CHIFFOLEAU (Floriane), *spellcheckTexts.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/spellcheckTexts.py> (visité le 19/04/2022).
- *textCorrection.Py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/textCorrection.py> (visité le 19/04/2022).
- Die Korrespondenz Der Constance de Salm (1767-1845). Inventar Des Fonds Salm Der Société Des Amis Du Vieux Toulon et de Sa Région Und Des Bestands Constance de Salm Im Archiv Schloss Dyck (Mitgliedsarchiv Der Vereinigten Adelsarchive Im Rheinland e. V.). Elektronische Edition*, 1^{er} avr. 2022, URL : <https://constance-de-salm.de> (visité le 11/04/2022).

Autres ressources numériques

- BARRUS (Tyler), *Pyspellchecker : Pure Python Spell Checker Based on Work by Peter Norvig*, version 0.6.3, URL : <https://github.com/barrust/pyspellchecker> (visité le 19/04/2022).
- CHIFFOLEAU (Floriane), *[Correspondance En Langue Française, XXe s.] SegmOnto*, 10 déc. 2021, URL : https://github.com/SegmOnto/examples/tree/main/sources/lettre_fr_XXe (visité le 07/04/2022).
- *Correspondence : Guidelines*, DAHN Project, 10 janv. 2022, URL : <https://github.com/FloChiff/DAHNProject/blob/master/Correspondence/Guidelines/Documentation-Correspondance.pdf> (visité le 07/04/2022).
 - *How to Do a Post-OCR Correction for TEXT Files*, DAHN Project, 8 avr. 2022, URL : <https://github.com/FloChiff/DAHNProject/blob/master/Project%20development/Documentation/Post-OCR%20correction%20for%20TEXT%20files.md> (visité le 11/04/2022).
 - *DAHN Project*, GitHub, URL : <https://github.com/FloChiff/DAHNProject> (visité le 05/04/2022).
 - *Few Tips for Reading the Text Files*, DAHN Project, URL : <https://github.com/FloChiff/DAHNProject/tree/master/Project%20development/Texts> (visité le 11/04/2022).
- CLÉRICE (Thibault), *HTRUC, HTR-United Catalog Tooling (Pronounced EuchTruc)*, version 0.0.1, nov. 2021, URL : <https://github.com/HTR-United/HTRUC> (visité le 20/05/2022).
- Docker Install [Installation d'eScriptorium]*, GitLab, URL : <https://gitlab.com/scripta/escriptorium/-/wikis/docker-install> (visité le 15/06/2022).
- Kraken [Documentation]*, Kraken, URL : <https://kraken.re/master/index.html> (visité le 28/04/2022).
- Kraken Models : Transcription Models*, GitLab Inria, URL : <https://gitlab.inria.fr/dh-projects/kraken-models/-/tree/master/transcription%20models> (visité le 28/04/2022).
- SpaCy : Industrial-strength Natural Language Processing in Python*, URL : <https://spacy.io/> (visité le 27/04/2022).

Études

- CHAGUÉ (Alix), *Création de modèles de transcription pour le projet LECTAUREP #1, Lectaurep : l'intelligence artificielle appliquée aux archives notariales*, URL : <https://lectaurep.hypotheses.org/475> (visité le 05/04/2022).

- CHAGUÉ (Alix), *Création de modèles de transcription pour le projet LECTAUREP #2*, Lectaurep : l'intelligence artificielle appliquée aux archives notariales, URL : <https://lectaurep.hypotheses.org/488> (visité le 05/04/2022).
- CHAGUÉ (Alix), CLÉRICE (Thibault) et ROMARY (Laurent), « HTR-United : Mutualisons La Vérité de Terrain ! », dans *DHNord2021 - Publier, Partager, Réutiliser Les Données de La Recherche : Les Data Papers et Leurs Enjeux*, Lille, 2021, URL : <https://hal.archives-ouvertes.fr/hal-03398740> (visité le 15/06/2022).
- GABAY (Simon), CAMPS (Jean-Baptiste), PINCHE (Ariane) et JAHAN (Claire), « SegmOnto : Common Vocabulary and Practices for Analysing the Layout of Manuscripts (and More) », dans *1st International Workshop on Computational Paleography*, Lausanne, 2021, URL : <https://hal.archives-ouvertes.fr/hal-03336528> (visité le 20/04/2022).
- GUYOTJEANNIN (Olivier), PYCKE (Jacques) et TOCK (Benoît-Michel), *Diplomatique médiévale*, 1993^e éd., Turnhout, 2006 (L'atelier du médiéviste, 2).
- JACQUOT (Olivier), *Transkribus / eScriptorium : transcrire, annoter et éditer numériquement des documents d'archives, ateliers à la BnF*, Carnet de la recherche à la Bibliothèque nationale de France, URL : <https://bnf.hypotheses.org/12575> (visité le 10/05/2022).
- « Leiden Conventions », dans *Wikipedia*, 2021, URL : https://en.wikipedia.org/w/index.php?title=Leiden_Conventions&oldid=1004624327 (visité le 05/05/2022).
- PINCHE (Ariane), « L'HTR : Présentation Des Problématiques Qui s'ouvrent Au Chercheur, Entre Particularité Du Document et Généralisation Du Modèle », dans *Conduite et Réalisation d'un Projet Informatique*, Cours de Master, Paris, École nationale des chartes, 2021.
- Séminaire ”Création de modèle(s) HTR pour les documents médiévaux en ancien français et moyen français entre le Xe-XIVe siècle” : compte-rendu de la séance n° 2, CREMMALAB, 2021, URL : <https://cremmalab.hypotheses.org/seminaire-creation-de-modeles-htr/compte-rendu-de-la-seance-n-2> (visité le 05/05/2022).
 - Séminaire ”Création de modèle(s) HTR pour les documents médiévaux en ancien français et moyen français entre le Xe-XIVe siècle” : compte-rendu de la séance n° 3, CREMMALAB, 2021, URL : <https://cremmalab.hypotheses.org/compte-rendu-seance-3> (visité le 13/06/2022).
- SOUVAY (Hippolyte), *La Correspondance de Constance de Salm (1767-1845) : Rapport de Stage*, rapport de stage de seconde année de master Humanités numériques et computationnelles, École nationale des chartes-Institut historique allemand à Paris, 2021.
- STOKES (Peter A.), KISSLING (Benjamin), STÖKL BEN EZRA (Daniel), TISSOT (Robin) et EL HASSANE (Gargem), « The eScriptorium VRE for Manuscript Cultures »,

Classics@ Journal (, 29 juil. 2021), URL : <https://classics-at.chs.harvard.edu/classics18-stokes-kiessling-stokl-ben-ezra-tissot-gargem/> (visité le 15/06/2022).

STUTZMANN (Dominique), « Paléographie statistique pour décrire, identifier, dater... Normaliser pour coopérer et aller plus loin ? », dans *Kodikologie und Paläographie im digitalen Zeitalter 2 - Codicology and Palaeography in the Digital Age 2*, dir. Franz Fischer, Christiane Fritze et Georg Vogeler, Norderstedt, 2011, t. 3, p. 247-277, URL : <https://kups.ub.uni-koeln.de/4353/> (visité le 08/01/2022).

Techniques et formats de conversion en mode texte, BnF - Site institutionnel, 2022, URL : <https://www.bnf.fr/fr/techniques-et-formats-de-conversion-en-mode-texte> (visité le 16/06/2022).

TEI : Text Encoding Initiative, URL : <https://tei-c.org/> (visité le 16/06/2022).

TORRES AGUILAR (Sergio), « e-NDP (Notre-Dame de Paris et son cloître) : 26 registres du chapitre de Notre-Dame de Paris datés du 14e-15e en latin (principalement) et français », dans *Transkribus / eScriptorium : transcrire, annoter et éditer numériquement des documents d'archives, ateliers à la BnF*, Paris, BnF, site François-Mitterrand, 2022.