

Contenu de la documentation

Présentation	5
Contexte	5
Objectifs	5
1 HTR : état des lieux	7
1.1 Principes généraux	7
1.2 La recherche à l'heure de l'HTR	9
1.2.1 Des finalités et des publics multiples	9
1.2.2 eScriptorium et Kraken : infrastructures et développements	11
1.3 Produire des modèles	12
1.3.1 Explorer de nouveaux types d'écriture	12
1.3.2 Méthodes d'acquisition et d'entraînement	13
1.3.3 Les sources et leurs problèmes	13
1.4 Partager les données	15
1.4.1 Partager les vérités de terrain et les modèles	15
1.4.2 Pour des données génériques	16
1.4.3 Exploiter les données : pour des solutions <i>open-source</i>	18
2 HTR : mise en œuvre	21
2.1 Enjeux et tâches préliminaires	21
2.1.1 Des sources écrites par plusieurs mains	21
2.1.2 Finalité : l'édition des lettres	22
2.1.3 Choisir des collections d'évaluation et identifier des mains	23
2.1.4 L'écriture personnelle de C. de Salm : un défi paléographique	25
2.1.5 Préparer le traitement d'un dossier	25
2.1.6 Transkribus ou eScriptorium ?	28
2.2 La segmentation	30
2.2.1 Principes et enjeux	30
2.2.2 Définir une structure de document idéale en vue de l'édition	32
2.2.3 Problèmes posés par l'espacement des lignes	34
2.2.4 Gérer la numérotation des lignes	35

2.2.5	Définir une ontologie des régions et des lignes	38
2.2.6	Résultats des entraînements	41
2.2.7	Contrôler la pertinence de la segmentation	42
2.3	La reconnaissance des caractères	43
2.3.1	Sélectionner des échantillons d'écriture et organiser les fichiers	44
2.3.2	Établir des normes de transcription	46
2.3.3	Transcription manuelle <i>versus</i> transcription automatique	48
2.3.4	Éliminer d'une transcription les lignes attestant des écritures parasites	49
2.3.5	Comparer les performances des modèles	50
2.3.6	Tenir un journal des résultats de tests et d'entraînements	52
2.3.7	Injecter les transcriptions manuelles dans les prédictions	53
2.4	La correction semi-automatisée	54
2.4.1	Trouver le bon compromis entre granularité et performance	55
2.4.2	Analyser les mots	56
2.4.3	Gérer les résolutions ambiguës	57
2.4.4	Élaborer et enrichir un nouveau dictionnaire de la langue française	57
3	Transformation TEI	63
3.1	Choisir un projet-modèle	64
3.2	Distribuer les fichiers de chaque lettre dans des dossiers	65
3.3	Récupérer les zones d'écritures pertinentes	66
3.4	Structurer l'encodage d'une lettre	68
3.5	Inscrire les métadonnées dans l'en-tête de la lettre	73
3.5.1	Décrire la lettre	73
3.5.2	Citer la notice de l'inventaire	74
3.5.3	Renseigner les données du projet	75
3.6	Finaliser l'encodage d'une lettre	75
Annexes		76
A	Transcriptions	
	de deux manuscrits autographes	
	de C. de Salm	79
A.1	Premier extrait	79
A.2	Second extrait	81
B	Normes de transcription	83
B.1	Accentuation	83
B.2	Majuscules et minuscules	83
B.3	Séparation des mots	83

B.4 Orthographe	83
B.5 Abréviations	84
B.6 Ponctuation	84
B.7 Passages biffés, palimpsestes	84
B.8 Passages illisibles	84
Glossaire	85
Acronymes	87
Bibliographie	89

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).

On s'appuiera pour cela sur les instruments et la documentation produits dans le cadre du projet Dispositif de soutien à l'Archivistique et aux Humanités Numériques (partenariat entre l'Inria (équipe ALMAnaCh), l'Université du Mans et l'EHESS) (DAHN), fondé sur l'édition de la correspondance de Paul d'Estournelles de Constant (1852-1924)¹.

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

Chapitre 1

Reconnaissance automatique des écritures manuscrites

Un état des lieux

La reconnaissance automatique des écritures manuscrites (ou *Handwritten Text Recognition* (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 d'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

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)).

identifiées et numérotées dans l'ordre de lecture (ce sans quoi la transcription produite serait inexploitable!) ;

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 transcris 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* (format XML pour la description des textes et de leur mise en page) (ALTO), maintenu par la Bibliothèque du Congrès et privilégié par la Bibliothèque nationale de France (BnF)².

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 apprentissage 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 (ou *accuracy*) 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 traitées 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

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) ; Id., « The eScriptorium VRE for Manuscript Cultures »...

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.

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 page transcrites à la main, il n'en va pas de même des modèles de segmentation, comme on aura l'occasion de le voir plus loin³.

1.2 La recherche à l'heure de l'HTR

Dans le cadre du projet Consortium Reconnaissance d'Écriture Manuscrite des Matériaux Anciens (Cremma)-Lab soutenu par le DIM MAP, le centre Jean-Mabillon (École nationale des chartes), en partenariat avec le LAMOP et le LabEX Hastec, a organisé les 23 et 24 juin 2022 un colloque intitulé *Documents anciens et reconnaissance automatique des écritures manuscrites*⁴.

Ce colloque a été l'occasion de rassembler une communauté scientifique représentant les pays du sud de l'Europe (France, Italie, Grèce, Portugal, Suisse) ainsi que quelques équipes nords-américaines autour des enjeux, des finalités, des problèmes et des solutions d'avenir de la reconnaissance automatique des écritures manuscrites.

Le compte-rendu analytique qui en est donné dans les pages qui suivent constitue un état de l'art des projets et des techniques appliquées à l'HTR.

1.2.1 Des finalités et des publics multiples

Les finalités de l'HTR sont multiples. Elles concernent aussi bien les scientifiques qu'un public élargi aux savants et aux curieux des sources écrites anciennes. Les projets *Crimes et châtiments*⁵ et *Lettres en lumières*⁶ ont démontré l'intérêt de l'HTR pour ouvrir la lecture des textes anciens en dehors du monde académique ou pour le développement de projets de transcription contributive.

3. Voir *infra* 2.2, p. 30.

4. Comité d'organisation : Ariane Pinche et Floriane Chiffolleau. Comité scientifique : Jean-Baptiste Camps, Alix Chagué, Thibault Clérice, Frédéric Duval, Vincent Jolivet, Benjamin Kiessling, Nicolas Perreau, Ariane Pinche, Laurent Romary, Peter Stokes. *Documents anciens et reconnaissance automatique des écritures manuscrites (HTR)*, colloque, 23 et 24 juin 2022, École nationale des chartes, Paris, programme et résumés, URL : <https://cremmalab.hypotheses.org/colloque-htr-programme> (visité le 03/05/2022)

5. Élodie Paupe, « Une cursive du XVIIe siècle », dans *Documents Anciens et HTR*, 2022.

6. Florian Fizaine et Édouard Bouyé, « Lettres en lumières », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

Pour le public scientifique, l’HTR est en mesure de rendre accessibles des données selon plusieurs modalités. Le projet POPP (Projet d’Océrisation des Recensements de la Population Parisienne) a montré comment elle permet de construire de vastes bases de données par l’extraction d’informations de recensements historiques⁷. Le projet Sofer Mahir a proposé une méthode pour l’établissement d’éditions critiques⁸, ce qui impose d’ajouter à la transcription des documents une étape de structuration de leur hiérarchie, les différents témoins d’un même texte affectant souvent des mises en page différentes. Les travaux de thèse de doctorat de Christophe Tufféry⁹ ont mis en évidence quant à eux un exemple de développement d’application visant à proposer, à partir de la transcription de carnets de fouilles archéologiques, des visualisations de ces données pour restituer l’histoire d’une fouille programmée.

Outre la mise à disposition des sources textuelles ou des données qu’elles contiennent, l’HTR offre des possibilités de traitement massif de ces données avec plusieurs types d’objectifs. Les *Expérimentations pour l’analyse automatique de sources chinoises anciennes*¹⁰ ont montré l’intérêt de l’HTR pour suivre l’utilisation de textes à travers les siècles. Dans les domaines épigraphique et paléographique également, les algorithmes de reconnaissance d’écriture peuvent servir d’outil à l’analyse des mots et des glyphes¹¹; l’analyse des erreurs de reconnaissance peut également être exploitée afin dégager des caractéristiques d’évolution des écritures¹².

Enfin le projet CHAMDOC a illustré le fait que l’HTR peut intervenir dans la préservation des langues écrites en péril, comme c’est le cas du cham ancien, langue véhiculaire utilisée dans des inscriptions gravées au Vietnam, du VIe au XVIIe siècle¹³.

7. Thomas Constum, « Reconnaissance et extraction d’informations dans des tableaux manuscrits historiques : vers une compréhension des recensements de Paris de l’entre-deux guerre », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

8. D. Stökl Ben Ezra, Lapin Hayim et Pavel Jablonski, « From HTR to Critical Edition : A Semi-Automatic Pipeline », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

9. Christophe Tufféry, « Retour d’expériences sur l’utilisation comparée de plusieurs de dispositifs de transcription numérique d’archives de fouilles archéologiques », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

10. Marie Bizaïs-Lillig et Chahan Vidal-Gorène, « Expérimentations pour l’analyse automatique de sources chinoises anciennes », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

11. Federico Boschetti et Tatiana Tommasi, « EpiSearch. Recognising Ancient Inscriptions in Epigraphic Manuscripts », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

12. Platanou Paraskevi, « HTR of Handwritten Paleographic Greek Text as a Function of Chronology », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

13. Anne-Valérie Schweyer, Jean-Christophe Burie et Tien Nam Nguyen, « Analyse, reconnaissance et indexation des manuscrits cham », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

1.2.2 eScriptorium et Kraken : infrastructures et développements

Le paysage des applications dédiées à l'HTR se partage depuis 2019 entre Transkribus (2016) et eScriptorium. Certains projets de recherche ont eu l'occasion de tester les deux applications¹⁴ et ainsi fait part de leurs expériences. L'entraînement de modèles HTR est un processus exigeant de très grandes capacités de calcul, et donc des infrastructures coûteuses. L'infrastructure Cremma ouvrira bientôt au public des institutions académiques partenaires une instance d'eScriptorium¹⁵ dotée de trois GPU (*Graphics Processing Unit* ou unité de traitement graphique), chacune représentant en moyenne un coût d'une dizaine de milliers d'euros. L'infrastructure FoNDUE de l'université de Genève bénéficie quant à elle de la puissance du superordinateur (HPC) de l'université, doté de 150 GPU. La recherche des meilleures performances des entraînements de modèles consiste généralement à trouver le bon équilibre entre temps de calcul par image et nombre de tâches lancées en parallèle¹⁶.

Les développements en cours de l'interface eScriptorium donneront lieu dans un avenir proche à une fonctionnalité de recherche des termes transcrits, à du balisage TEI basique, à la possibilité d'annotation graphique des pages, à l'alignement automatique d'un texte existant sur une image et à l'intégration de l'ordre des lignes dans l'entraînement des modèles de segmentation. Quant à l'application Kraken, sur laquelle se fonde l'interface eScriptorium, sa dernière version stable (4) propose de nouvelles bibliothèques d'entraînement et une meilleure accessibilité de son API. Elle affiche en outre une amélioration des performances pour les modèles de reconnaissance d'écriture, la reconstruction de lacunes et une nouvelle technologie de segmentation des régions et des lignes d'écriture (*layout analysis*) : grâce à l'utilisation de la technologie Transformers, la détection de l'orientation des lignes est désormais plus robuste, et il devient possible de segmenter des lignes qui se croisent. Cette innovation est néanmoins très exigeante sur le plan de l'infrastructure et renchérit le coût technologique de l'entraînement de modèles¹⁷.

14. Élina Leblanc et Pauline Jacsont, « De Transkribus à eScriptorium : retour(s) d'expérience sur l'usage d'outils d'HTR appliqués à un corpus d'imprimés espagnols du XIXe siècle », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022 ; É. Paupe, « Une cursive du XVIIe siècle »...

15. Elsa Marguin-Hamon, « Discours d'ouverture et présentation des projets CREMMA et CREMALAB », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

16. Simon Gabay et Pierre Künzli, « FoNDUE - A Lightweight HTR Infrastructure for Geneva », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

17. B. Kiessling et P. A. Stokes, « New Developments in Kraken and eScriptorium », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

1.3 Produire des modèles

1.3.1 Explorer de nouveaux types d'écriture

Les projets de recherche présentés à l'occasion du colloque s'emploient à étendre la zone de compétence des modèles HTR dans les domaines les plus variés. Tandis que les écritures livresques médiévales sont de mieux en mieux couvertes du XI^e au XV^e siècle par les modèles produits dans le cadre du projet Cremma : Arabica, Bicerin et bientôt Cortado¹⁸, le projet e-NDP s'emploie à travers les sources du chapitre de Notre-Dame de Paris à entraîner des modèles pour des écritures nouvelles : *cursiva*, *textualis*, prégothique, semihybride¹⁹. En effet les écritures cursives font actuellement partie des fronts pionniers de l'entraînement des modèles, que ce soit pour le Moyen Âge, le XVII^e siècle²⁰, ou dans le contexte de projet diachroniques comme l'étude des archives inquisitoriales portugaises pour la période allant 1536 à 1821²¹, ou encore des sources très contemporaines comme les carnets de fouilles archéologiques du XX^e siècle²². Comme l'a montré le projet TraPrInq (*Transcribing the court records of the Portuguese Inquisition, 1536-1821*), l'entraînement de ces modèles doit parfois relever le défi de la variété paléographique, des mélanges de langues et de types d'écriture (latine, arabe, hébraïque) ; de la variété aussi des types de documents dont découle des mises en pages hétérogènes, de la variété des supports également, susceptible d'infléchir l'efficacité des entraînements²³. D'autres projets tentent de couvrir une diachronie encore plus longue, comme la création d'un corpus de fictions littéraires allant du XI^e siècle à nos jours²⁴. Enfin, les projets HTR s'étendent également en direction des écritures non latines, comme l'ont illustré des présentations du projet CHAMDOC²⁵ ainsi que les expérimentations pour l'analyse automatique de sources chinoises anciennes²⁶.

18. Jean-Baptiste Camps et Ariane Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

19. Sergio Torres Aguilar et Vincent Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

20. É. Paupe, « Une cursive du XVII^e siècle »...

21. Hervé Baudry, « Les archives inquisitoriales (Portugal) sous HTR : le projet TraPrInq (*Transcribing the court records of the Portuguese Inquisition, 1536-1821*) », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

22. C. Tufféry, « Retour d'expériences sur l'utilisation comparée de plusieurs de dispositifs de transcription numérique d'archives de fouilles archéologiques »...

23. H. Baudry, « Les archives inquisitoriales (Portugal) sous HTR : le projet TraPrInq (*Transcribing the court records of the Portuguese Inquisition, 1536-1821*) »...

24. J.B. Camps et A. Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts »...

25. A.V. Schweyer, J.C. Burie et Tien Nam Nguyen, « Analyse, reconnaissance et indexation des manuscrits cham »...

26. M. Bizais-Lillig et C. Vidal-Gorène, « Expérimentations pour l'analyse automatique de sources chinoises anciennes »...

1.3.2 Méthodes d'acquisition et d'entraînement

En dehors des projets très pionniers comme ceux portant sur les écritures extrême-orientales, l'époque est révolue de la production de vérités de terrain *ex nihilo* à chaque nouveau projet, comme l'a rappelé Vincent Jolivet²⁷. La méthode désormais privilégiée consiste à repérer des modèles existants (*transfer learning*). Dans ce contexte, le partage des données d'entraînement et des modèles devient essentiel. Les meilleurs modèles identifiés sont ensuite affinés (*fine-tuning*) grâce à de nouvelles vérités de terrain. Tout le problème consiste à savoir de quelle quantité de données d'entraînement l'on aura besoin pour atteindre le score nécessaire, et quelle est la meilleure méthode pour optimiser cette étape de la production d'un modèle affiné. La réponse ne peut être qu'empirique tant les ressources disponibles sont variables (les paléographes sont rares!) et les gains d'acuité attendus de ces entraînements (qui dépendent de la finalité de chaque projet).

Le projet ETSO (*Estilometría aplicada al Teatro del Siglo de Oro*) a également montré que l'acquisition des données d'entraînement peut procéder par une autre voie que la transcription manuelle, à savoir la récupération d'éditions de textes existantes et leur alignement avec la reproduction photographique de page manuscrites, une tâche d'ores- et-déjà possible grâce à la fonction Text2IMage de Transkribus et bientôt développée par eScriptorium²⁸.

Les techniques d'apprentissage constituent bien souvent la clé du succès pour l'entraînement d'un modèle robuste. La personnalisation de ces techniques, possible avec l'application Kraken, a joué un rôle important pour un projet comme « Crimes et châtiments »²⁹. La modification du paramètre de la vitesse d'apprentissage (le paramètre -r de Kraken) s'est en effet répercutée sur l'acuité du modèle HTR pour l'écriture cursive du XVIIe siècle de l'ancien Évêché de Bâle.

1.3.3 Les sources et leurs problèmes

Les types d'écriture

L'entraînement des modèles HTR implique souvent de trouver des solutions adaptées à la complexité des sources, qu'elle soit de nature matérielle avec la qualité des reproductions photographiques³⁰, paléographique (variété des types d'écriture, diversité des systèmes de signes) ou qu'elle ait trait à la mise en page des documents.

27. S. Torres Aguilar et V. Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation »...

28. Álvaro Cuéllar, « Un modèle ouvert pour la reconnaissance automatique des manuscrits du théâtre espagnol du Siècle d'Or », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

29. **paupeCursive17eSiecle2022**.

30. É. Leblanc et P. Jacsont, « De Transkribus à eScriptorium : retour(s) d'expérience sur l'usage d'outils d'HTR appliqués à un corpus d'imprimés espagnols du XIXe siècle »...

Le projet e-NDP, confronté à une assez large variété d'écritures gothiques, s'emploie à développer une méthode de classification automatique des écritures. À partir de modèles entraînés pour chaque type d'écriture, cette méthode permet d'évaluer automatiquement le type d'écriture d'un document selon les performances des différents modèles. Une fois le type d'écriture automatiquement identifié, le modèle adéquat peut être appliqué³¹.

L'équipe travaillant sur les sources chinoises anciennes a quant à elle proposé une méthode spécifique pour l'entraînement à partir de données lacunaires³². En effet le chinois ancien comporte environ 54 000 caractères, dont seuls 4000 à 5000 sont représentés dans le jeu de données retenu : des textes xylographiés de la Chine impériale (50 images). Or tous les caractères attestés ne sont pas référencés dans le système Unicode et il peut exister plusieurs glyphs valant pour le même caractère. L'ampleur du système de signes combinée à la taille réduite du jeu de données a pour conséquence que de très nombreux signes contenus dans le jeu de validation sont absents du jeu d'entraînement. Une méthode développée pour l'étude de l'écriture arabe a ainsi été mise en place : une fois réalisé un premier entraînement classique de modèle HTR, les glyphs du jeu d'entraînement sont utilisés pour forger de faux glyphs à partir d'un vaste corpus de textes glané sur le web. Cette méthode permet d'affiner le modèle HTR en l'aident à reconnaître des séquences de caractères. La méthode a prouvé son efficacité en faisant descendre le taux d'erreur par caractère (CER pour *character error rate*) à 14% seulement pour les glyphs absents du jeu de données primaire.

Les types de mise en page

L'étude des écritures chinoises anciennes soulève en outre des problèmes ayant trait à la segmentation des lignes. L'écriture verticale des sources xylographiques étudiées présente deux types de configuration des caractères : des colonnes simples et des colonnes doubles. Dans le second cas, de grands caractères servent de rubrique, un peu comme les titres chapteautant plusieurs colonnes dans la presse écrite. Cette succession verticale de grand caractères centrés et de petits caractères sur deux colonnes contrarie l'entraînement d'un modèle de segmentation fondé sur la principe de la ligne de base (*baseline*) ; pour répondre à ce problème, la technique de la boîte englobante (*bouding-box*) a été appliquée afin d'identifier les types de colonnes (double ou simple) et de pouvoir ensuite traiter correctement chaque région du texte de manière adaptée³³. L'analyse de mises en page complexes a également été abordée par l'équipe du Geniza Lab³⁴ qui travaille sur une

31. S. Torres Aguilar et V. Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation »...

32. M. Bizais-Lillig et C. Vidal-Gorène, « Expérimentations pour l'analyse automatique de sources chinoises anciennes »...

33. *Ibid.*

34. D. Stökl Ben Ezra, Marina Rustow et Deborah Witty, « Segmentation Mode for Archival Documents with Highly Complex Layout », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

très vaste typologie de documents médiévaux en hébreu, judéo-arabe et araméen. Certains documents analysés présentent en effet un corps de texte disposé en lignes horizontales et une glose marginale disposée de manière giratoire autour du premier ; des marges sont en outre écrites la tête en bas. Deux méthodes ont été envisagées pour l'entraînement de modèles de segmentation :

1. Entraîner huit modèles différents avec une rotation de l'image à 45 degrés entre chaque entraînement pour reconnaître chaque orientation de texte ;
2. Entraîner un seul modèle (sans rotation) avec une annotation (corrigée manuellement) des régions et des lignes d'écriture selon leur orientation, chaque orientation étant annoté de manière propre.

C'est finalement cette seconde option qui a été retenue comme la plus efficace.

1.4 Partager les données

Pour exprimer leur plein potentiel les avancées technologiques de l'HTR supposent, encore plus que l'échange des bons procédés et des méthodes innovantes, le partage des données. Les infrastructures d'entraînement coûtent cher, mais c'est aussi le cas de la production de vérités de terrain qui exige des compétences rares (paléographiques, linguistiques) et beaucoup de temps pour les mettre en œuvre soigneusement.

1.4.1 Partager les vérités de terrain et les modèles

Le projet Cremma-Lab contribue à cet effort de partage selon deux voies :

1. La production de vérités de terrain pour les écritures latines allant du Moyen Âge au XXe siècle (21000 lignes de transcription ont été publiées de manière ouverte) ;
2. Le développement du catalogue HTR-United pour l'identification des vérités de terrain disponibles en ligne³⁵³⁶.

Le catalogue HTR-United référence d'ores-et-déjà (au 30 juin 2022) les productions de 46 projets, comptabilisant plus de 380 000 lignes. Il propose un moteur de recherche permettant d'interroger les types d'écriture, les langues, et la chronologie des sources. Un standard de description des métadonnées permet à tout projet de signaler ses propres vérités de terrain afin d'être correctement référencé dans le catalogue.

35. J.B. Camps et A. Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts »...

36. Alix Chagué et Thibault Clérice, « Sharing HTR Datasets with Standardized Metadata : The HTR-United Initiative », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

1.4.2 Pour des données génériques

La réutilisation des vérités de terrain implique d'en apprécier la proximité avec des sources données. Comment choisir les données d'entraînement permettant d'obtenir les meilleurs scores de son propre projet ? Les expérimentations du projet Cremma-Lab permettent d'établir qu'il n'est pas la peine de trop spécialiser ses données d'entraînement par langue ou par type de texte pour obtenir de bons résultats de reconnaissance³⁷. Mais obtenir les meilleurs scores possibles n'est pas sans inconvénient. Si ces sources sont trop similaires, l'entraînement devient synonyme d'hyper-spécialisation et les scores risquent de chuter pour des sources nouvelles. Le modèle Bicerin a ainsi pu être « amélioré » par l'ajout de données très différentes et ce malgré une diminution de ses scores d'acuité (*accuracy*). La robustesse d'un modèle HTR (on parle également de sa générnicité, de sa souplesse ou de sa plasticité) signifie sa capacité à reconnaître des types d'écritures hétérogènes. Faire varier les langues et les genres littéraires ou les types de texte contribue également à cette robustesse dans la mesure où les modèles HTR s'appuient sur un modèle de langue susceptible d'influencer les prédictions. Trouver le meilleur équilibre entre générnicité et acuité particulière est au fond une équation très empirique, de même qu'apprécier la proximité ou la distance entre deux écritures. Ainsi la notion de « hors-domaine », employée pour l'évaluation d'un modèle HTR sur des données étrangères aux données d'entraînement, recouvre une très grande variété de situations selon la diversité des sources étudiées, comme cela a été évoqué au sujet du projet e-NDP³⁸.

Les présentations ont montré combien la diversité des projets et de leurs finalités est irréductible à une méthodologie unique. Pourtant, nombre de projets partagent les mêmes enjeux et dépendent du partage des mêmes données d'entraînement. C'est pourquoi le projet Cremma-Lab³⁹ propose des réflexions méthodologiques sur les protocoles de transcriptions des corpus, dans un souci d'harmonisation des pratiques. Cet effort passe par la rédaction d'un guide de transcription (*guidelines*). Les règles communes de transcription ainsi proposées pourront permettre l'entraînement de modèles HTR plus robustes dans la mesure où ces entraînements reposeront sur des transcriptions plus homogènes. Il s'agit en l'occurrence d'appliquer quelques principes simples : ne pas imiter les formes de lettres lorsqu'elles se rapportent à un même caractère (allographes) et conserver les abréviations (dont le développement dépend souvent du contexte, de la langue écrite ou *scripta*). Pour les abréviations et autres caractères spéciaux, il s'agit de suivre une table de caractères Unicode de référence, l'outil ChocoMufin développé par HTR-United permettant le contrôle des caractères employés et ainsi d'éviter que des caractères Unicode

37. J.B. Camps et A. Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts »...

38. S. Torres Aguilar et V. Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation »...

39. J.B. Camps et A. Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts »...

différents ne soient mobilisés de manière concurrente pour la résolution d'un signe graphique équivalent.

Du côté de l'analyse de la mise en page et des lignes d'écriture, une autre initiative dans le sens de l'harmonisation des vérités de terrain a pris forme avec le projet SegmOnto : A Controlled Vocabulary to Describe the Layout of Pages (SegmOnto)⁴⁰. D'abord fondé sur l'étude des manuscrits médiévaux et des imprimés anciens, SegmOnto propose une ontologie complète pour la description des régions et des lignes qui se veut suffisamment générique pour répondre à l'ensemble des besoins d'annotation, quelque soit le type de source ancienne ou contemporaine étudié. Ces concepts relèvent par conséquent de la description matérielle plutôt que de la fonction sémantique des régions d'écriture (la notion de « marge » a par exemple été préférée à celle de glose et celle d'objet « graphique » à la notion de décoration ou d'image). La liste des zones est la suivante :

- CustomZone
- DamageZone
- DigitizationArtifactZone
- DropCapitalZone
- GraphicZone
- MainZone
- MarginTextZone
- MusicZone
- NumberingZone
- QuireMarksZone
- RunningTitleZone
- SealZone
- StampZone
- TableZone
- TitlePageZone

Au niveau des lignes, l'onotologie est plus resserée :

- CustomLine
- DefaultLine
- DropCapitalLine
- HeadingLine
- InterlinearLine
- MusicLine

Outre des catégories génériques recommandées, une syntaxe a été établie afin d'introduire de la modularité ; on pourra par exemple définir des colonnes de la façon suivante : par *MainZone :columnA*, *MainZone :columnB*. En outre, les catégories *CustomZone* et

⁴⁰ S. Gabay, A. Pinche et Kelly Christensen, « SegmOnto – A Controlled Vocabulary to Describe Historical Textual Sources », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

CustomLine permettent, de la manière décrite dans l'exemple précédent, de définir des catégories personnalisées.

1.4.3 Exploiter les données : pour des solutions *open-source*

Exploiter les données : pour des solutions open-source

Atteindre un certain degré d'harmonisation des pratiques ne signifie pas nécessairement contraindre les finalités des projets. Au contraire, la conciliation de cette double exigence (harmonisation, finalités multiples) est rendue possible par le recours à une méthode de travail séquentielle : une chaîne de traitement clairement structurée en étapes successives qui procèdent indépendamment les unes des autres. Entraîner des modèles d'HTR déjà performants implique d'apporter beaucoup de données d'entraînement pour gagner, en définitive, très peu d'acuité. Au-delà d'une certaine performance acquise, il peut donc être pertinent de répondre aux problèmes de l'HTR (les fautes dans les prédictions) autrement que par l'entraînement de modèles encore plus performants, à savoir en corrigeant les prédictions dans une phase ultérieure de la chaîne de traitement⁴¹.

Ces étapes ultérieures peuvent être multiples : lemmatisation des mots, normalisation des graphies, annotation des entités nommées, etc. Leur utilité est propre aux finalités de chaque projet : produire des modèles HTR robustes ne nécessite aucun traitement particulier, car le partage du modèle est une fin en soi⁴²; en revanche, rendre un corpus de textes interrogable sur plusieurs siècles suppose de lui appliquer des solutions d'uniformisation éditoriale⁴³; établir une édition critique peut encore supposer des traitements particuliers, comme séparer les mots (*tokenization*) lorsque la langue de la source atteste beaucoup d'agglutinations⁴⁴, etc. En bout de chaîne, le développement de modèles de publication, comme a pu l'illustrer le projet Gallic(orpor)a (projet financé par Huma-Num et le BnF DataLab) (Gallic(orpor)a)⁴⁵, sera à n'en pas douter l'un des enjeux importants pour les projets HTR dans les prochaines années.

Dans ce contexte, la mise en œuvre par la communauté scientifique de solutions applicatives indépendantes et complémentaires les unes des autres comporte plusieurs types de bénéfices. En ne faisant pas dépendre les projets de recherche d'un logiciel unique (potentiellement défaillant, comme toute application), elle est un gage de sécurité et de pérennité de ces projets. Le développement d'applications *open-source* rend de plus chaque

41. S. Torres Aguilar et V. Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation »...

42. J.B. Camps et A. Pinche, « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts »...

43. M. Bizais-Lillig et C. Vidal-Gorène, « Expérimentations pour l'analyse automatique de sources chinoises anciennes »... ; S. Torres Aguilar et V. Jolivet, « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation »...

44. D. Stökl Ben Ezra, L. Hayim et P. Jablonski, « From HTR to Critical Edition : A Semi-Automatic Pipeline »...

45. S. Gabay, A. Pinche et K. Christensen, « SegmOnto – A Controlled Vocabulary to Describe Historical Textual Sources »...

fonctionnalité perfectible de manière contributive au rythme des besoins de la communauté ou d'un projet singulier. Enfin et surtout, traiter les sources par étapes successives permet l'archivage des états intermédiaires du travail et donc le contrôle de ces étapes par des projets ultérieurs ou, encore une fois, le partage des données entre des projets n'ayant pas les mêmes finalités⁴⁶.

46. ^campsCremmaLabProjectsTranscription2022.

Chapitre 2

Mettre en œuvre l’HTR pour la correspondance de C. de Salm

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.

2.1 Enjeux et tâches préliminaires

2.1.1 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 Répertoires (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)¹ ?

Les résultats probants obtenus par le projet Lectaurep en suivant l'option d'entraînement d'un modèle mixte² 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 Salm³ ;
2. Reprendre un modèle générique entraîné dans le cadre du projet Lectaurep pour en évaluer les performances.

2.1.2 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 question⁴.

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%⁵.

1. A. 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).

2. 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).

3. 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.

4. 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.

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

2.1.3 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.

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 six⁶.

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.

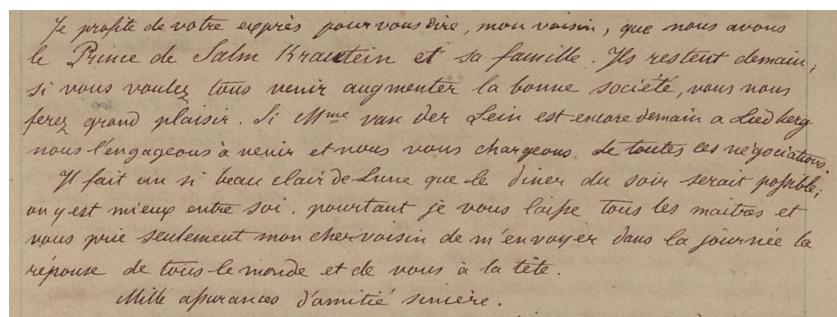


FIGURE 2.1 – Première main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-02_0065.jpg).

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

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

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 distinguées.

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

6. 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).

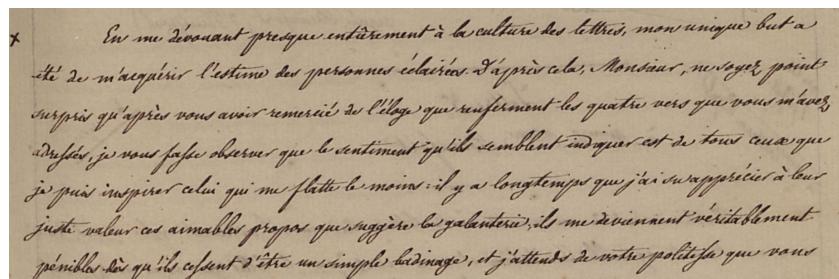


FIGURE 2.2 – Deuxième main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-01_0030.jpg).

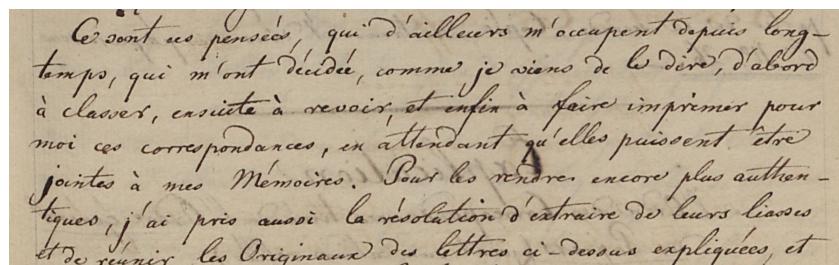


FIGURE 2.3 – Troisième main principale des recueils de la deuxième copie (LAI, détail du cliché CdS02_Konv002-01_0006.jpg).

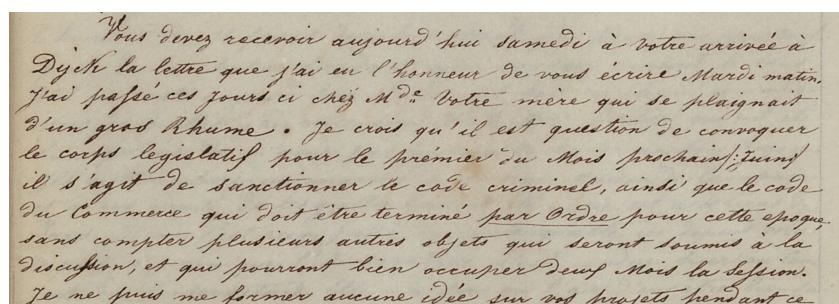


FIGURE 2.4 – Première main de la correspondance Martini (LAI, détail du cliché CdS02_Konv019_0002.jpg).

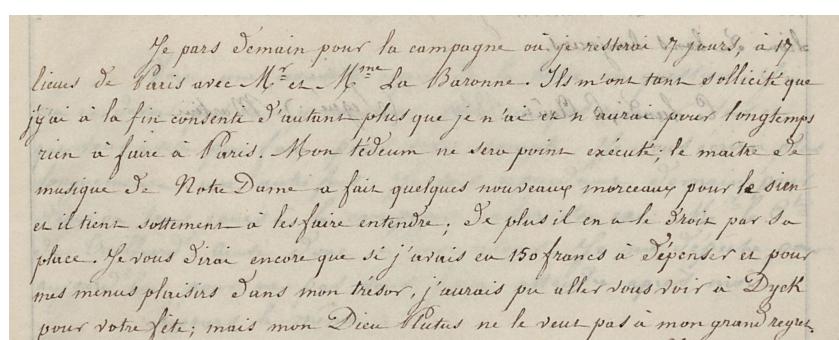


FIGURE 2.5 – Seconde main de la correspondance Martini (notice CdS/19/036-037, URL : <https://constance-de-salm.de/archiv/#/document/10504>).

problèmes de transparence qui font ressortir au recto l'encre du verso (un problème assez présent dans la correspondance Martini).

2.1.4 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.

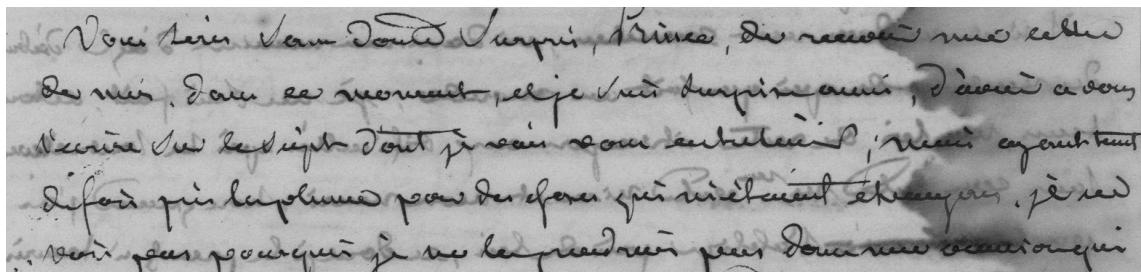


FIGURE 2.6 – Écriture autographe de C. de Salm (notice C11/S92/047-049, URL : <https://constance-de-salm.de/archiv/#/document/765> (visité le 13/06/2022), transcription *infra*, p. 81).

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%⁷.

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 lettres⁸.

2.1.5 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> ; il permet d'afficher ces données dans un fichier au format *JavaScript Object Notation* (format standard de représentation de données structurées) (Json), en donnant la liste des images concernées.

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

8. Voir *infra*, A, p. 79.

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 2.7. Cette photographie contient la fin d'une lettre⁹, le début d'une autre¹⁰, et entre les deux une lettre non inventoriée, dont le titre se situe en haut de la page de droite.

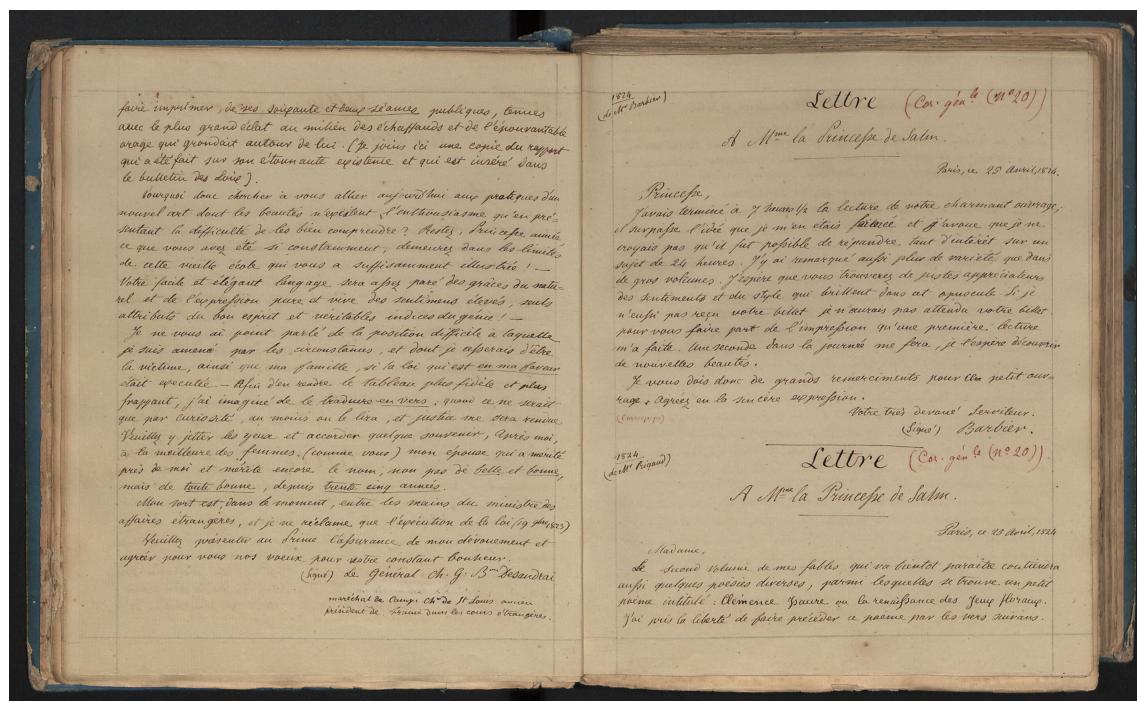


FIGURE 2.7 – Image contenant trois lettres dont l'une n'est pas inventoriée (cliché : CdS02_Konv002-03_0058.jpg)

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

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

10. *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 consacré¹¹. 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 python¹² 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 notice¹³.

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 2.7 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. 2.2.7).

11. 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.

12. Id., *donneesImages.py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesImages.py>.

13. 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*, 2.2.7, p. 42)

2.1.6 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 applications¹⁴. 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 avancé¹⁵, 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 Kraken¹⁶), promeut les principes de la science ouverte de multiples manières (développement *open-source*, respects de standards des formats numériques, ouverture des

14. 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).

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

16. *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 eScrip-torium encourage (politiquement) la publication et le partage des vérités de terrain (qui sont les véritable s données brutes d'entraînement) ainsi que des modèles eux-mêmes¹⁷.

On peut ajouter à cette considération sur le 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 incompatibles 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 données¹⁸.

Pour l'ensemble de ces raisons, nous avons opté pour l'utilisation d'eScrip-torium 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 permet en toute théorie de mener l'intégralité des tâches que l'on a expérimentées avec une

17. A. Chagué, T. 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).

18. P. A. Stokes, B. Kiessling, D. Stökl Ben Ezra, *et al.*, « The eScrip-torium VRE for Manuscript Cultures »...

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 locale¹⁹, faisant appel aux seules ressources d'un ordinateur portable, à savoir sans serveur ni carte graphique externe²⁰. 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 Cremma rendrait possible.

2.2 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.

2.2.1 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*)²¹. 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 et Kraken reconnaissent les lignes de base des écritures (*baselines*) et peuvent ainsi repérer

19. 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).

20. 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.

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

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.

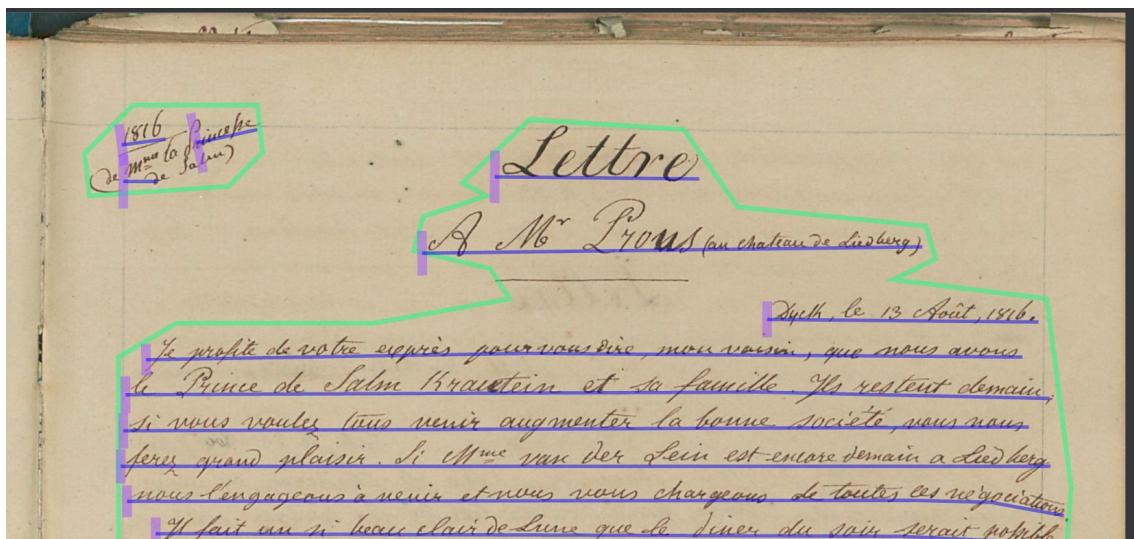


FIGURE 2.8 – Segmentation par le modèle par défaut de Kraken (LAI, détail du cliché CdS02_Konv002-02_0065.jpg).

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 propre 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.

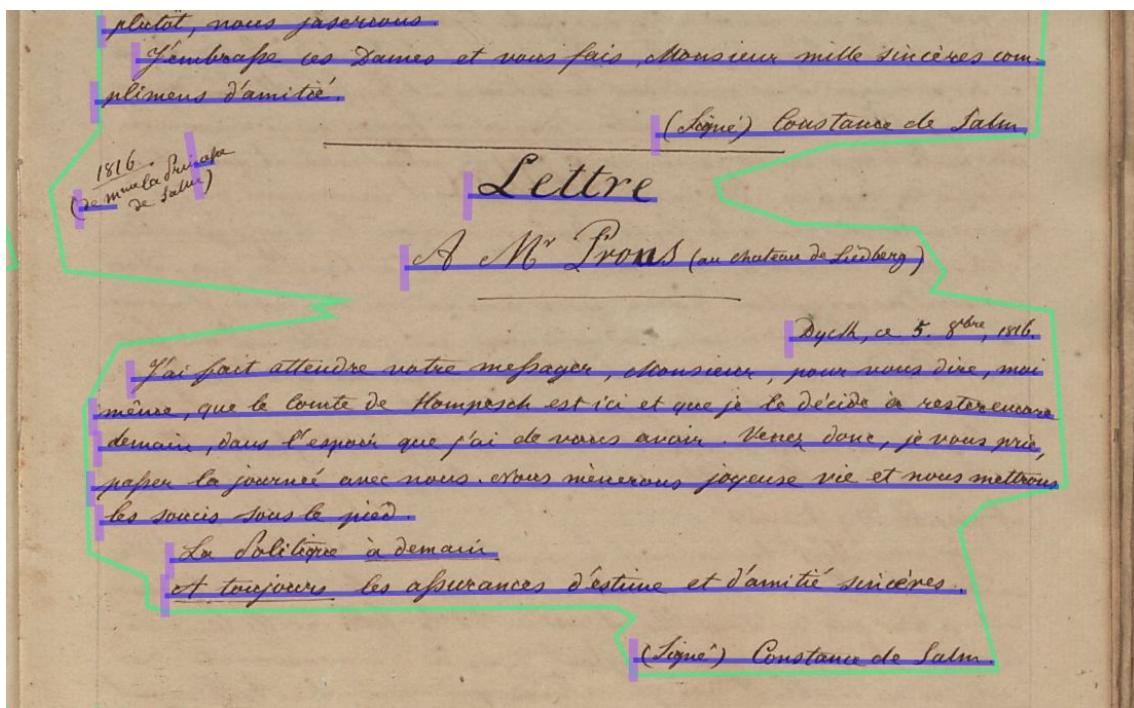


FIGURE 2.9 – Segmentation par le modèle par défaut de Kraken (notice : Cds/02_2/073, URL : <https://constance-de-salm.de/archiv/#/document/8855>).

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 doit 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.

2.2.2 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 an-

noter (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 DAHN²². En lien avec ces propositions d’encodage, F. Chiffoleau a en outre élaboré une ontologie des régions d’écriture pour les correspondances en langue française pour le xx^e siècle²³, dans le cadre du projet SegmOnto²⁴. 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 2022²⁵ 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.*

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.

22. F. Chiffoleau, *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).

23. 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).

24. S. Gabay, J.B. Camps, A. 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).

25. On participé à cette discussion Anne Baillot, Floriane Chiffoleau, Pauline Spychala, Evan Virevialle et moi-même.

2.2.3 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 2.8, p. 31). L'exemple illustré par la figure 2.10 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).

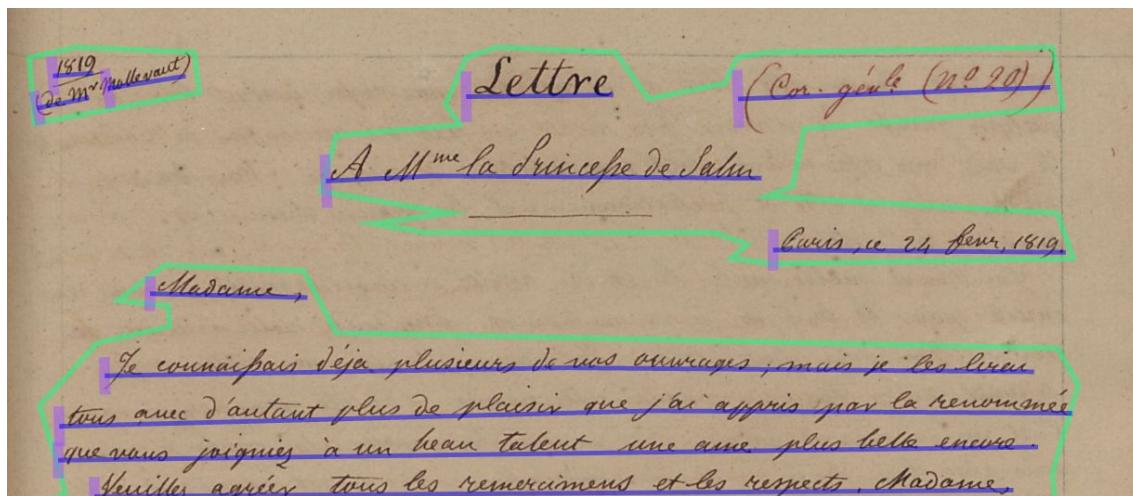


FIGURE 2.10 – Mise en page aérée au début d'une lettre (LAI, détail du cliché CdS02_Konv002-02_0193.jpg).

En revanche, dans l'exemple illustré par la figure 2.11, 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 celui illustré par la figure 2.12, 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.

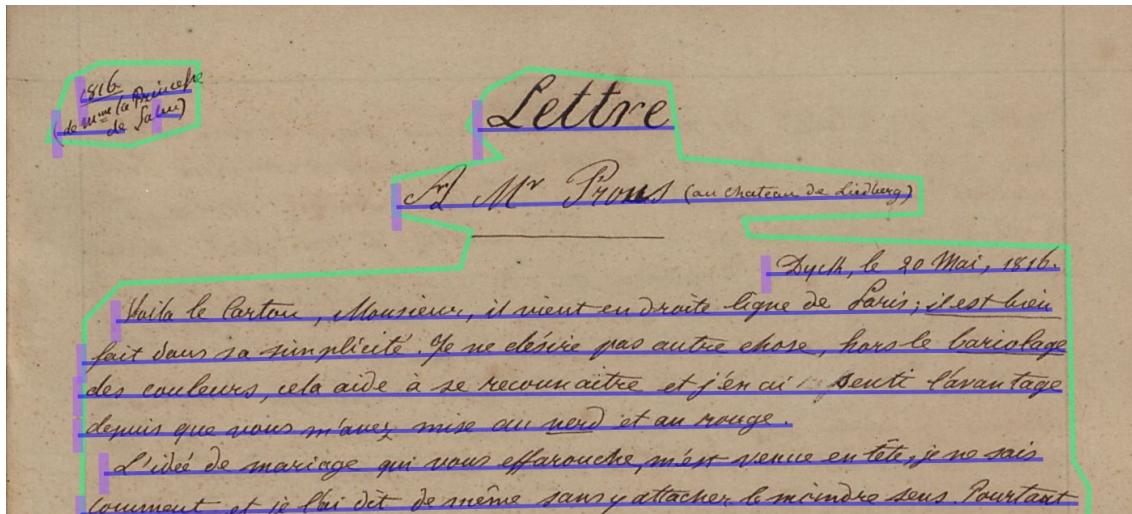


FIGURE 2.11 – Mise en page resserrée au début d'une lettre (LAI, détail du cliché CdS02_Konv002-02_0046.jpg).

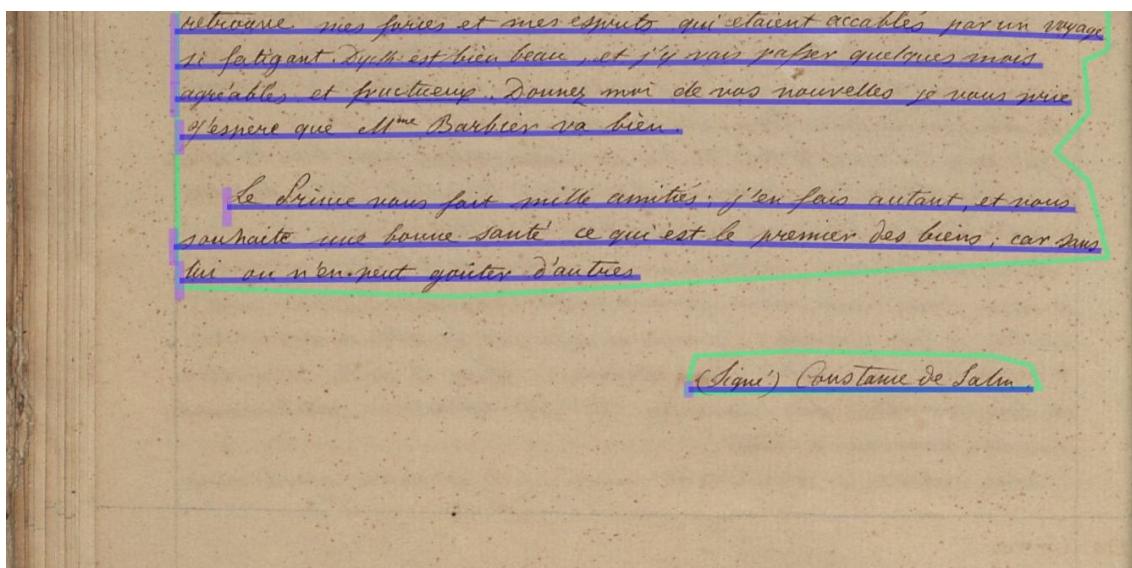


FIGURE 2.12 – Mise en page aérée en fin de lettre (notice : Cds/02_3/070-071, URL : <https://constance-de-salm.de/archiv/#/document/8907>).

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.

2.2.4 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 transcrrites 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 2.13, issue du vol. 2 de la seconde copie²⁶, 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 rechargement 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) 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

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

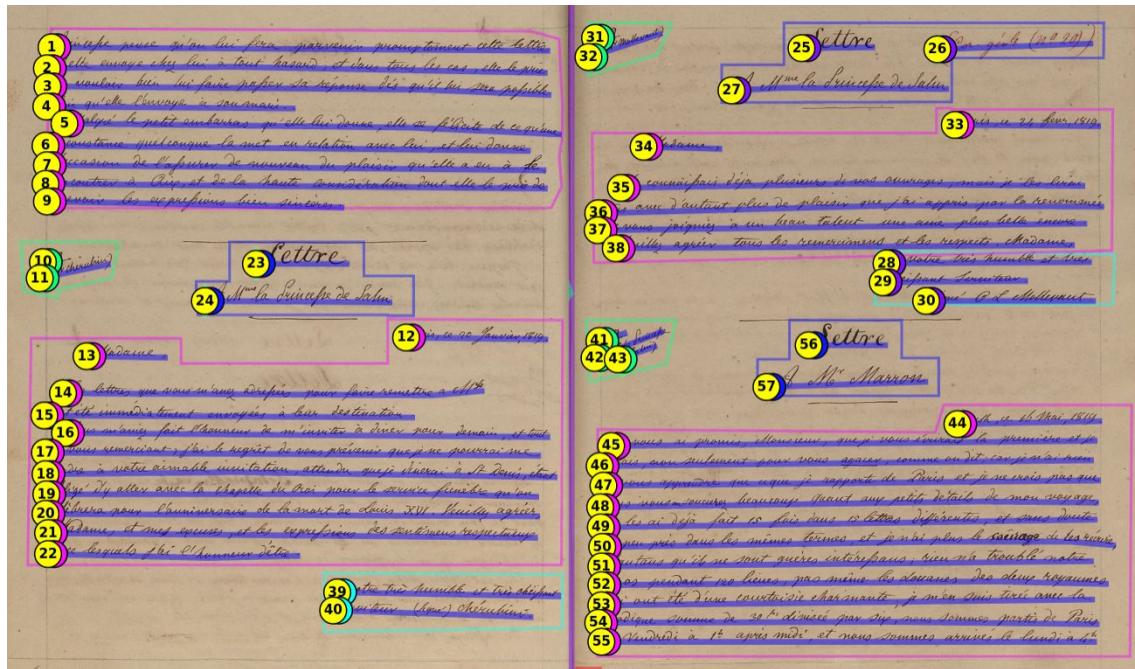


FIGURE 2.13 – Exemple de numérotation à partir d'une segmentation distinguant *opener* et *closer* (cliché CdS02_Konv002-02_0193.jpg).

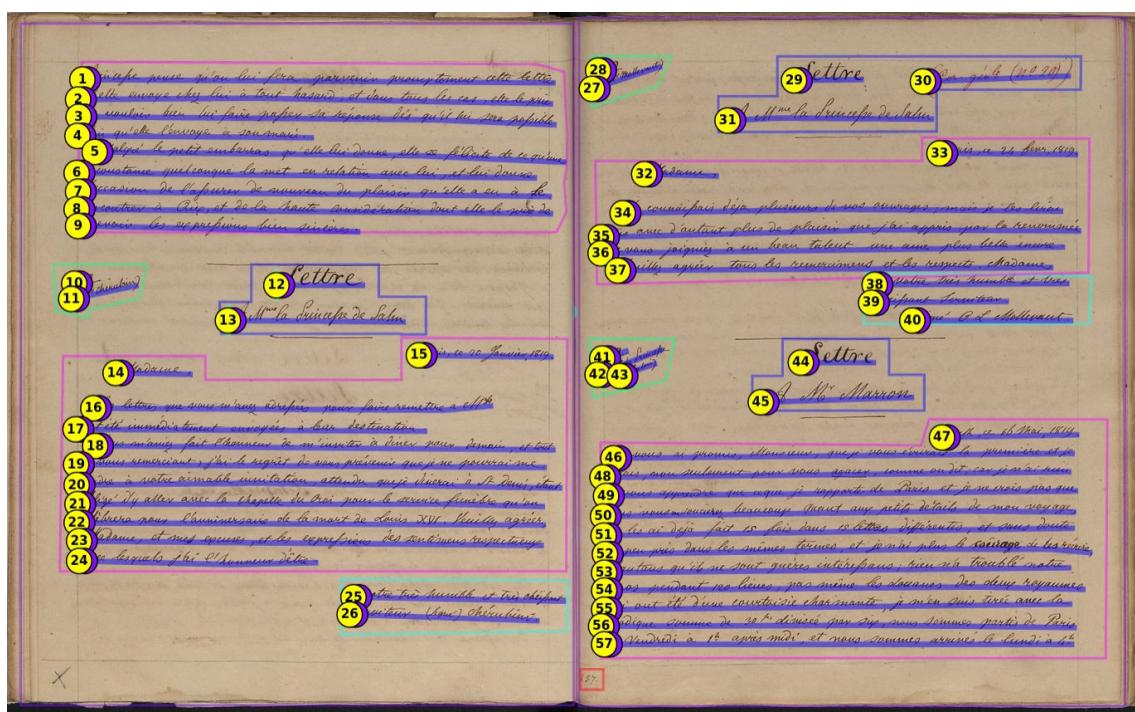


FIGURE 2.14 – Exemple de numérotation avec lignes liées à une région *page* (cliché CdS02_Konv002-02_0193.jpg).

problèmes de numérotation.

La solution finalement apparue comme la seule possible consistait à simplifier la définition des régions d'écriture. Dans la figure 2.15, 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.

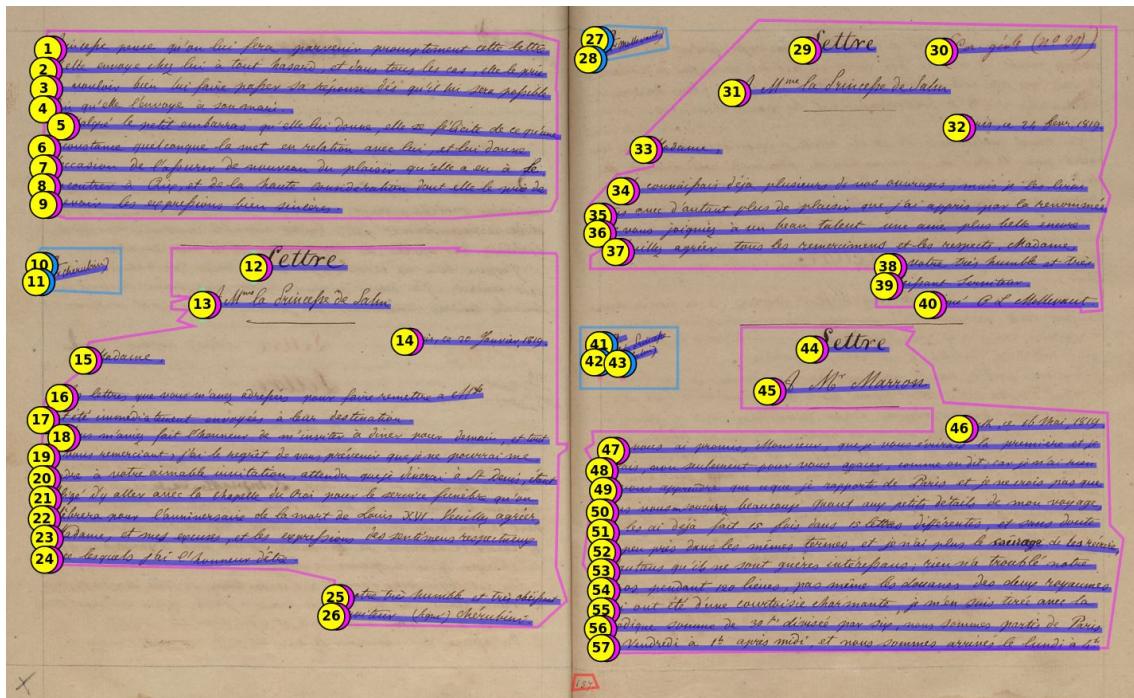


FIGURE 2.15 – Exemple de numérotation sans région de *page* ni *opener* (cliché CdS02_Konv002-02_0193.jpg).

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.

2.2.5 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 page 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 HTRUC²⁷, 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/#/document/10517> (visited on 06/21/2022) en haut de la page de gauche).

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 2.16 que la mention entre parenthèses *Cor(respondance) gén(éra)le (n° 20)* que l'on ici isolée dans une région à part a reçu le

27. 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).

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.

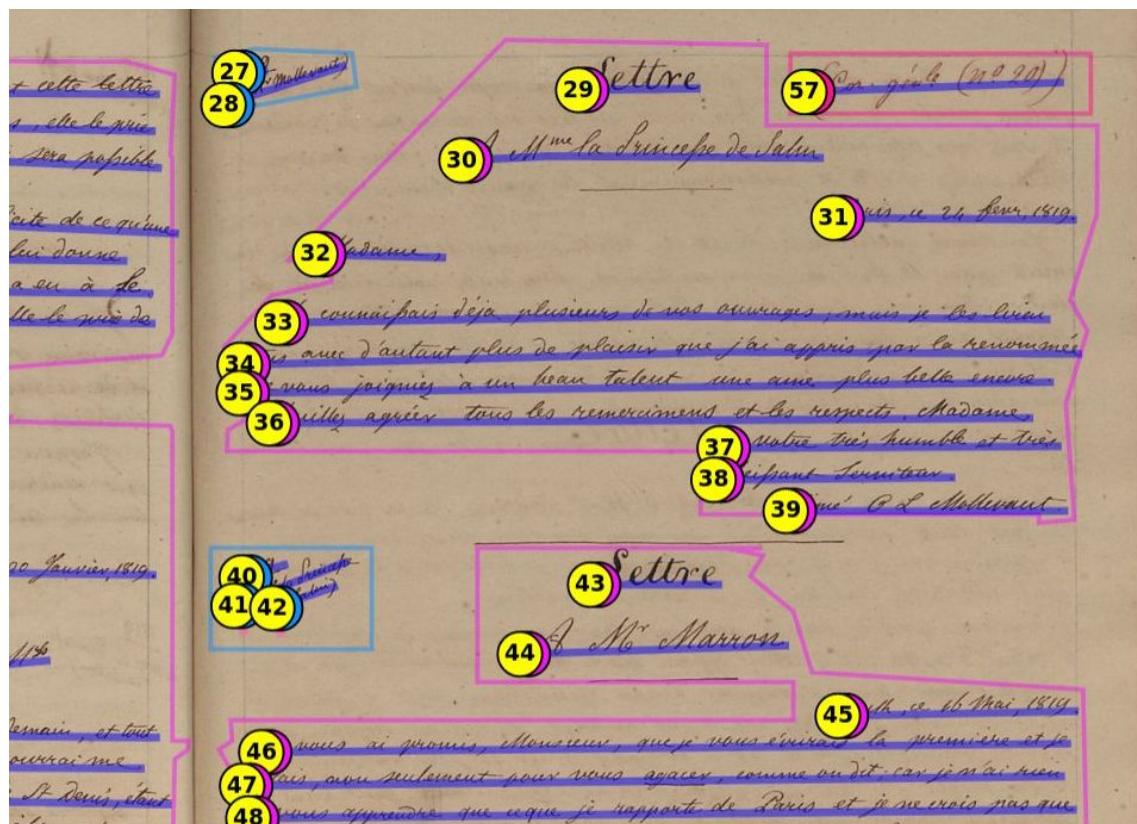


FIGURE 2.16 – Problème de numérotation lié à une région de type *MarginTextZone* (cliqué : CdS02_Konv002-02_0193.jpg).

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 répercutee 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.

2.2.6 Résultats des entraînements

Les principes de segmentation que l'on vient de définir ont été appliqués à un premier lots de 123 doubles pages issues des 2^e et 3^e volumes des recueils de copies²⁸. Les pages ont été initialement segmentées avec le modèle par défaut de Kraken, puis corrigées et annotées manuellement dans l'interface eScriptorium²⁹. Une fois ces corrections effectuées, les pages ont été exportées au format ALTO et un modèle de segmentation affinant le modèle par défaut a été entraîné avec l'application Kraken³⁰.

La figure ?? montre l'application du modèle entraîné à une nouvelle double page et permet de constater les performances du nouveau modèle³¹.

Celui-ci est parvenu à définir de manière correcte les manchettes (entourées en bleu) et le corps des lettres, réunissant toutes les parties, du titre à la signature, en leur attribuant la bonne typologie de région. En revanche le petit numéro situé en bas de la page de droite, bien que repéré par le segmenteur, n'a pas reçu la bonne annotation, puisqu'il a été catégorisé de la même manière que la manchette. Concernant les lignes d'écriture, on remarque qu'elles ont été globalement bien identifiées. Chose importante, toutes les lignes que l'on souhaitait voir annotées comme **DefaultLine** l'ont été correctement.

En revanche, ce sont les autres types de lignes que le segmenteur n'a pas correctement repéré ni annoté. Seules les lignes de titres ont été correctement annotées **HeadingLine**, mais on remarque qu'elles n'ont pas été bien repérées, car leur ligne de base est parfois découpée en plusieurs tronçons là où l'on attendrait une seule ligne horizontale. Les lignes de date, de signature, de post-scriptum ou de manchette ont quant à elle été très mal ou pas du tout repérées.

Bien que très imparfait, ce résultat n'en constitue pas moins un relatif succès de l'entraînement, dans la mesure où la segmentation produite est tout à fait exploitable. Elle peut être corrigée manuellement beaucoup moins de temps qu'une annotation manuelle

28. *CdS/02_2/001-369 : Correspondance générale, seconde copie, 2^e volume, 1815-1821...* ; *CdS/02_3/001-334 : Correspondance générale, seconde copie, 3^e volume, 1822-1828...*

29. Les fichiers exportés sont les 123 premiers contenus dans le dossier <https://github.com/sbiay/CdS-edition/tree/main/htr/entrainements/train-seg-Konv002>.

30. La commande utilisée, qui indique l'architecture d'entraînement qui nous a été conseillée par A. Pinche, est la suivante :

```
ketos segtrain -bl entrainements/train-seg-Konv002/*.xml -i ./blla.mlmodel --resize both --f alto -o ./sortie/ --device cpu --augment --schedule reduceonplateau -s "[1,1800,0,3 Cr7,7,64,2,2 Gn32 Cr3,3,128,2,2 Gn32 Cr3,3,128 Gn32 Cr3,3,256 Gn32 Cr3,3,256 Gn32 Lbx32 Lby32 Cr1,1,32 Gn32 Lby32 Lbx32]" --merge-baselines DefaultLine:default --merge-regions MainZone:text.
```

31. Le modèle est déposé à l'adresse suivante : <https://github.com/sbiay/CdS-edition/blob/main/htr/modeles-seg/copie-deux-04.mlmodel>.

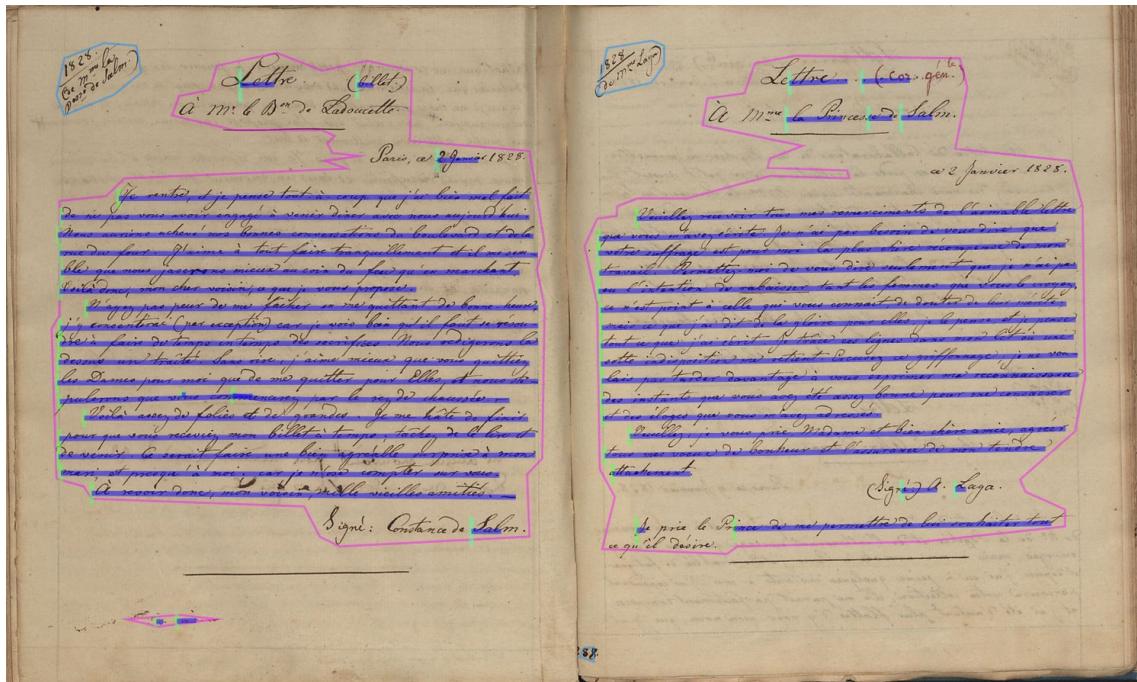


FIGURE 2.17 – Résultat de l’entraînement d’un modèle de segmentation sur 123 doubles pages de recueils.

consécutive à une segmentation avec modèle par défaut.

On a tenté de confirmé ces résultats encourageants en apportant de nouvelles pages au jeu d’entraînement. Mais contrairement à ce que l’on espérait, avec 61 doubles pages supplémentaires corrigées à la main³², les résultats se sont révélés moins bons que lors du premier entraînement, car les lignes de titre ont été encore moins bien reconnu que précédemment.

La préparation des données et les entraînements eux-mêmes demandant beaucoup de temps et d’effort, il a fallu renoncer à de nouvelles expérimentations pour ne pas desservir les autres étapes de la chaîne de traitement.

2.2.7 Contrôler la pertinence de la segmentation

Le script python écrit pour permettre l’importation sélective des images inventoriées³³ 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

32. Il s’agit de l’intégralité des fichiers contenus dans le dossier <https://github.com/sbiay/Cds-edition/tree/main/htr/entrainements/train-seg-Konv002>

33. S. Biay, *donneesImages.py...*

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.

2.3 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 démarche³⁴.

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 copies³⁵. 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 suivre³⁶.

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

35. 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>).

36. Id., *Tester et entraîner 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. 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.

2.3.1 Sélectionner des échantillons d’écriture et organiser les fichiers

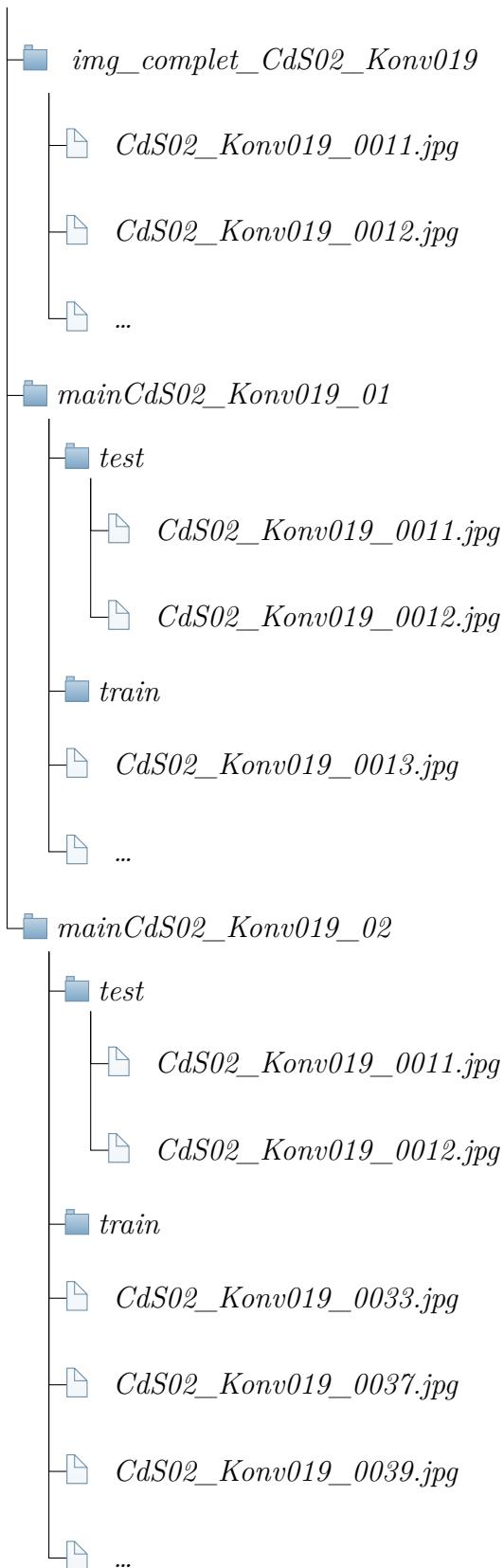
Entraîner des modèles à reconnaître les écritures de plusieurs mains différentes suppose un regard attentif aux variations paléographiques, mais aussi une grande rigueur de gestion des fichiers et de leurs données, car il s’agit d’abord de classer par type d’écriture les reproductions photographiques 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.

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 loin³⁷.

37. Cf. *infra* 2.3.6, p. 52.

entraînements

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 transcriptions³⁸.

2.3.2 Établir des normes de transcription

Il faut évoquer brièvement ici les principes généraux de la transcription des textes, les normes détaillées étant reportées en annexe³⁹.

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 lettres⁴⁰), soit comme une transcription déjà interprétative de la source qui uniformise les allographes et restitue les abréviations⁴¹. En réalité, aucune de ces options ne peut être appliquée de

38. 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*

39. Cf. B, p. 83.

40. Cette méthode de transcription est généralement dénommée allographétique ; cf. **stutzmannPaleographieStatistiquePour2011a**

41. C'est le cas de la transcription dite diplomatique ; cf. Olivier Guyotjeannin, Jacques Pycke et

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 chapitre⁴². 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 abrégé⁴³.

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*⁴⁴. 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 prononcés par la communauté scientifique constituée autour du projet Cremma et qui correspondent de manière tendancielle au concept de transcription *graphématisque* traduit et expliqué par D. Stutzmann⁴⁵.

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 difficulté 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

Benoît-Michel Tock, *Diplomatique médiévale*, 1993^e éd., Turnhout, 2006 (L'atelier du médiéviste, 2)

42. S. 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.

43. 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.

44. 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° 3, CREMMALAB, 2021, URL : <https://cremmalab.hypotheses.org/compte-rendu-seance-3> (visité le 13/06/2022).

45. stutzmannPaleographieStatistiquePour2011a.

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 DAHN⁴⁶. Plutôt que d'introduire des caractères tels que £, €, etc. dans les transcriptions, ce qui les éloignerait d'une transcription de type graphématisée et limiterait les possibilités de réutilisation éventuelles de ces transcriptions, on a préféré appliquer les conventions préconisées par la convention de Leyde⁴⁷, retenues dans le cadre du Cremma⁴⁸ (cf. p.84).

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'encodage⁴⁹.

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

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.⁵⁰

46. 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).

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

48. 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).

49. Voir *infra* p. 41.

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

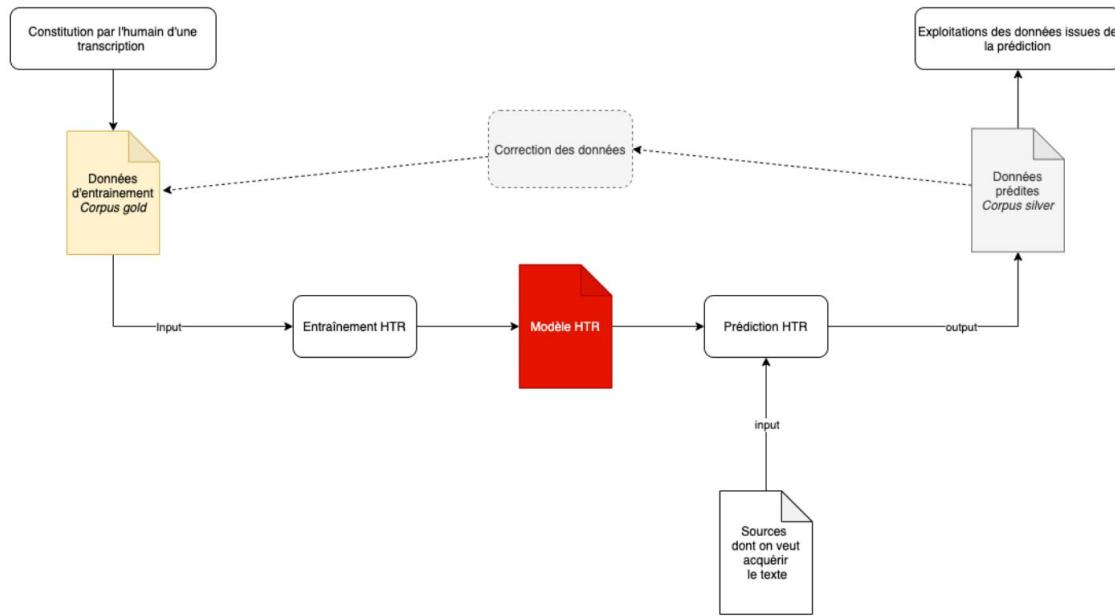


FIGURE 2.18 – Schéma d’entraînement récursif appliquée à la transcription (PINCHE (Ariane), « L’HTR : présentation des problématiques qui s’ouvrent au chercheur, entre particularité du document et généralisation du modèle », dans *Conduite et Réalisation d’un Projet Informatique*, Cours de Master, Paris, École nationale des chartes, 2021).

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 parfait, 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 scripturaux (notamment l’accentuation des mots) imposait un contrôle visuel plus intense pour corriger une transcription déjà faite que pour produire cette transcription.

2.3.4 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 travail⁵¹. 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.

2.3.5 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 commande⁵² (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. Souvay⁵³, 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.

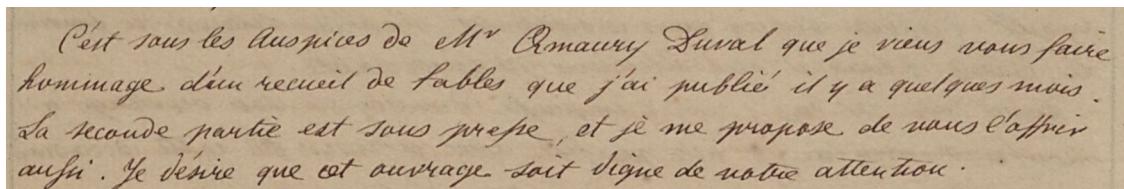


FIGURE 2.19 – Copie d’une lettre de Pierre-Augustin Rigaud à C. de Salm, le 13 avril 1824.

Voici la prédiction correspondante :

cet ju le gusaires de Mr touaauz 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'affir
auissi. Je dhrre qe cet ouvrage sait digne de votre atenson⁵⁴

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

51. S. Biay, *supprLignesVides.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/supprLignesVides.py>.

52. *Kraken [Documentation]*...

53. H. Souvay, *La correspondance de Constance de Salm (1767-1845) : rapport de stage...*

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

à partir d'un premier modèle mixte dans le cadre du même projet, qui atteignait quant à lui une acuité de 80,42%⁵⁵ :

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 performances⁵⁶.

En procédant à la constitution d'une vérité de terrain d'une dizaine de pages⁵⁷ pour 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 lettres⁵⁸ : 98,68%
- 3e main de la seconde copie des lettres⁵⁹ : 96,31%
- 1re main de la correspondance Martini⁶⁰ : 97,88%
- 2e main de la correspondance Martini⁶¹ : 96,27%

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

55. 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.

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

57. 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.

58. Dénommée mainCdS02_Konv002_01.

59. Dénommée mainCdS02_Konv002_03.

60. Dénommée mainCdS02_Konv019_01.

61. Dénommée mainCdS02_Konv019_02.

C'est sous les Auspices de Mr Amaury Duval que je viens vous faire hommage d'**em** recueil de **I**ables que j'ai publié il y a quelques mois La seconde partie est sous presse, et je me propose de vous l'**assir** 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.

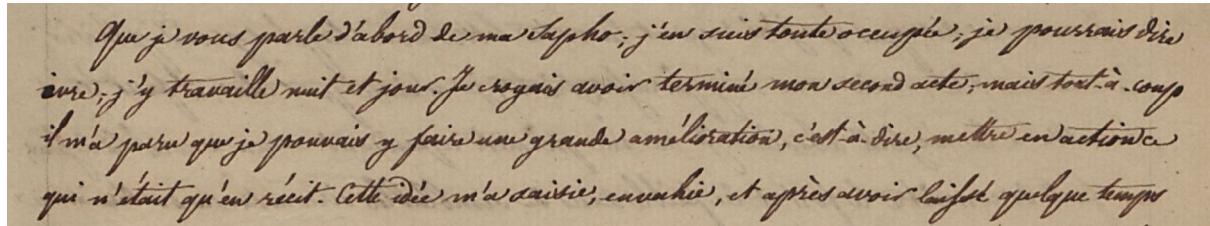


FIGURE 2.20 – Copie d'une lettre adressée par C. de Salm à Jean-François Thurot, le 21 février 1794.

Que j vous parle d'abord le ma Saphe ; j'en suis toute occupé, je pourrais dire
évre 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 tempps⁶²

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).

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

Les performances du nouveau modèle, dénommé `cds_lectcm_04_mains_01`⁶³, 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 résultats⁶⁴.

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

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

64. S. Biay, *journalReconn.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/journalReconn.py>.

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 modèle⁶⁵.

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.

2.3.7 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

65. Le fichier contenant ces données se trouve à l’adresse suivante : <https://github.com/sbiay/CdS-edition/blob/main/htr/mains/mains.\gls{json}>.

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 python⁶⁶ 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.

2.4 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.

Toutefois, 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.

Mais automatiser la correction des prédictions requiert de la prudence. Il faut naturellement veiller à ne pas remplacer involontairement des prédictions justes au regard des normes d'édition retenues. En l'occurrence, ces normes suivent de près les usages scribes, la graphie des mots ne saurait donc être uniformisée par la correction automatisée : la notion de justesse est élargie aux variations graphiques de chaque scribe. De ce point de vue, l'automatisation des corrections peut s'avérer précieuse (on a parlé de tamis) 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

⁶⁶ Id., *injectTranscript.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/injectTranscript.py>.

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 prédiction par l'éditeur, mais pas qu'elle produise un texte ayant le statut de vérité de terrain, encore moins 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 vers le format TEI, où une correction manuelle approfondie du texte est nécessaire pour son établissement définitif.

La démarche explicitée dans la documentation du projet DAHN⁶⁷ a été suivie et enrichie de quelques développements appliqués aux scripts de ce projet⁶⁸.

2.4.1 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. En effet, 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 : faut-il accepter la proposition de correction, la modifier, ou ne rien faire ? Les propositions de correction doivent donc être les moins nombreuses et les plus pertinentes possibles, afin de ne pas gaspiller le temps de l'éditeur ; 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'écriture, ainsi que les corrections validées pour les pages précédemment traitées⁶⁹.

En résumé, la correction automatisée se concentre sur l'orthographe des mots. Elle

67. 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).

68. 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>. 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).

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

ne traite pas la ponctuation. De plus, elle considère qu'une forme présente dans les vérités de terrain, dans les corrections déjà validées 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 ; par exemple, dans la prédiction *Dans vu siècle où tous les talens...*, la prédiction erronée *vu* pour *un* ne sera pas signalée, car le mot *vu* est attesté ailleurs.

2.4.2 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*, il est encore capable d'associer la forme *pubiès* à un mot de la famille de *publier* (deux fautes), mais en revanche, il n'est pas capable de faire cette association pour la forme *pubiès*, qui par rapport aux formes de la famille de *publier* qu'il connaît comporte au moins trois fautes.

Afin de faciliter la correction des dictionnaires générés par le script pour chaque page (ce sont ces dictionnaires au format Json 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 textes⁷⁰. 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 entraînait la création d'une faute : *naturellement* devenait *natureellement*. Le script a donc été perfectionné afin de procéder à l'application des corrections ligne par ligne et mot par mot⁷¹.

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'origine⁷².

70. S. Biay et F. Chiffoleau, *textCorrection.py*, 6 avr. 2022, URL : <https://github.com/sbiay/Cds-edition/blob/main/htr/py/textCorrection.py>.

71. 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).

72. L'injection des transcriptions manuelles en lieu et place des prédictions (cf. ci-dessus, 2.3.7, p. 53) 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 dès lors plus possible de s'appuyer sur l'identité des indentations pour repérer les lignes de textes à remplacer. Il devenait ainsi obligatoire de s'appuyer

2.4.3 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 prédiction 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*).

On a procédé dans un premier temps selon une méthode qui neutralisait les corrections ambiguës : *cele* était intégré à la liste globale des corrections avec une absence de résolution 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 écrite⁷³ 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 intervenir sur la correction à effectuer.

2.4.4 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'application⁷⁴, ces dictionnaires ont été élaborés à partir de la col-

sur la hiérarchie de l'arbre XML pour appliquer ces corrections.

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

74. T. Barrus, *Pyspellchecker...*

lecte massive de formes de mots parmi les ressources du site OpenSubtitles⁷⁵, 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, vinit, 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 des prédictions. On a ainsi fait l'expérience que la forme *ette*, qui n'existe pas en français mais est attestée pas moins de 36 fois dans le dictionnaire francophone de Pyspellchecker, a de fait été considérée comme juste, passant à travers les mailles du filet de la correction automatique.

Pour remédier à ce problème, on a fait appel au répertoire de textes numérisés de la ressource en ligne Frantext⁷⁶. Frantext est un corpus numérique développé par l'Analyse et Traitement Informatique de la Langue Française (Centre National de la Recherche Scientifique-Université de Lorraine) (ATILF), dont la dernière version (mise à jour en janvier 2022) comptabilise 5555 références et 264 millions de mots. Les textes y sont proposés dans un encodage TEI très simple et avec une lemmatisation de chaque forme. Plus intéressé par les formes des mots que par leur lemme, on a pu facilement récupérer

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

76. *Frantext*, URL : <https://www.frantext.fr/> (visité le 22/06/2022).

toutes les formes contenues dans une sélection de textes pour construire un dictionnaire au format Json.

Dans un premier temps, il s'agissait de procéder à une sélection des textes les plus pertinents dans le vaste corpus de Frantext. Trois critères complémentaires ont été retenus pour établir une première collection de cinquante textes. Premièrement, tous les textes devaient être dans le domaine public afin d'éviter toute difficulté liée à la réutilisation des données. Deuxièmement, les textes devaient avoir été publiés grossso-modo dans la période d'activité de C. de Salm : la fourchette 1780-1850 a ainsi été retenue comme critère de sélection. Troisièmement, les textes les plus riches en mots ont été privilégiés, afin de permettre la constitution du dictionnaire le plus complet possible à partir d'un petit nombre de textes exportés⁷⁷.

Au sein de la collection de cinquante textes qui a résulté de ces critères⁷⁸, on a procédé à une seconde sélection. On a ainsi retenu quelques textes parmi les plus volumineux pour disposer rapidement d'un dictionnaire important, des éditions de correspondance et de la littérature épistolaire pour rejoindre le genre des sources du projet (*Les liaisons dangereuses* offrant en outre l'avantage d'une date haute, à la différence de la plupart des titres de la collection), et un ouvrage historique volumineux susceptible de contenir de nombreux noms propres, à savoir les *Mémoires d'outre-tombe*.

Voici la liste complète des références sélectionnées, par ordre alphabétique des noms d'auteurs :

1. BALZAC Honoré, 1844, *Le Lys dans la vallée*, roman ;
2. CHATEAUBRIAND François-René de, 1848, *Mémoires d'outre-tombe* ;
3. CHODERLOS DE LACLOS Pierre-Ambroise-François, 1782, *Les Liaisons dangereuses*, roman ;
4. GAUTIER Théophile, 1843, *Voyage en Espagne*, récit de voyage ;
5. GUÉRIN Eugénie de, 1847, *Lettres (1831-1847)*, correspondance ;
6. HUGO Victor, Marie, 1848, *Correspondance : t. 1 : 1814-1848*, correspondance ;
7. SUE Eugène, 1843, *Les Mystères de Paris*, roman ;
8. SUE Eugène, 1845, *Le Juif errant*, roman.

Une fois les fichiers de ces références téléchargés et stockés, leur contenu a été récupéré par un script Python⁷⁹. Celui-ci ouvre chaque fichier, en récupère toutes les formes

77. Car bien que les textes soient librement téléchargeables dès lors que l'on dispose d'un compte d'utilisateur académique (celui de la Bibliothèque de l'École des chartes en l'occurrence), l'application pose une limite au volume de données téléchargeables, une limite vite atteinte avec un petit nombre de textes volumineux.

78. La liste est consultable à l'adresse suivante : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/dicos/frantext/recherche-frantext.csv>.

79. S. Biay, *dictGenerateur.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/dictGenerateur.py>.

sémantiques, compte le nombre d'occurrences de chaque forme et les inscrit dans un fichier au format Json qui constitue le dictionnaire final⁸⁰. Dans le même temps, ce script complète un tableau dans lequel sont listés les textes pris en compte (dès lors que ceux-ci existent dans la collection de cinquante textes que l'on a décrite plus haut). Il est donc des plus aisés de télécharger de nouveaux textes de la collection et de relancer le script : l'enrichissement du dictionnaire se fera en même temps que la mise à jour des sources qu'il contient.

Une fois ce dictionnaire constitué, il était intéressant de le comparer au dictionnaire francophone par défaut de Pyspellchecker. Le nouveau dictionnaire est riche de presque 60 000 formes, parmi lesquelles il faut compter des noms propres, des doublons de forme dans la mesure où la casse a été respectée (les mots peuvent donc être attestés deux fois, avec ou sans majuscule), ainsi que les signes de ponctuation (ils ne représentent naturellement qu'une poignée d'entrées). Sur ces 60 000 formes, moins de 5 000 sont absentes de l'ancien dictionnaire (et l'on peut soupçonner que de nombreux noms propres ou graphies anciennes sont impliqués) ; en revanche presque 800 000 formes de l'ancien dictionnaire sont absentes du nouveau. Si l'on suppose un instant que le dictionnaire que l'on vient de constituer couvre très largement les besoins de notre projet (son utilisation l'a confirmé), on peut en déduire que le dictionnaire de Pyspellchecker contenait, du point de vue du projet, 94% de formes parasites. On peut donc estimer que le gain de performance pour l'exécution de l'analyse automatique des mots a été énorme.

Mais une analyse automatique des transcriptions appuyée sur ce seul dictionnaire reste insuffisante et la nécessité d'un dictionnaire propre à la correspondance de C. de Salm s'est trouvée confirmée à l'usage. En effet, notre nouveau dictionnaire généraliste ignore la variété des usages scribaux concernant l'accentuation des mots (en particulier l'usage de l'accent aigu), et il ignore bien entendu les noms propres des familiers de C. de Salm.

En outre, une différence de conception sépare le nouveau dictionnaire généraliste du dictionnaire des formes relevées dans la correspondance. Pour les besoins de la correction des transcriptions automatiques, il était utile, dans le dictionnaire de la correspondance, de considérer comme une forme unique deux mots reliés par une élision (*j'ai*, *l'on*, etc.). En effet, la reconnaissance automatique d'écriture peut facilement ne pas voir l'élision d'un mot (il transcrit alors *jai*, *lon*), d'où le besoin de corriger fréquemment ce problème en prenant les deux mots comme un tout. Dans le nouveau dictionnaire généraliste, les élisions sont traitées comme deux formes distinctes, et donc séparées. Cette discordance entre dictionnaire généraliste et dictionnaire de la correspondance avait pour inconvénient que, si la correction automatique de *effvoi* en *effroi* fonctionnait bien, en revanche la forme *l'effvoi* avec son élision ne pouvait pas être automatiquement mise en relation avec le mot

80. Le fichier se trouve à l'adresse suivante : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/dicos/general.\gls{json}>.

effroi. Il y a été remédié sans modifier la conception spécifique de chaque dictionnaire, mais simplement en ignorant les élisions au moment de l'analyse des mots par le script SpellcheckTexts⁸¹.

81. S. Biay et F. Chiffolleau, *spellcheckTexts.py*...

Chapitre 3

Transformation des transcriptions automatiques en édition XML-TEI

L'étape finale de la chaîne de traitement consiste à transformer les transcriptions automatiques en édition nativement numérique. Le format d'édition privilégié est la TEI.

La *Text Encoding Initiative*, communauté scientifique qui donne son nom au format TEI, propose des principes pour l'encodage des textes ainsi qu'un format standard pour l'échange des données textuelles¹. Depuis le début des années 2000, sa syntaxe est fondée sur le langage Extensible Markup Language (XML). Fondée aux États-Unis, cette communauté est internationale. Sa vocation à permettre l'encodage de tous les types de textes est à l'origine des innombrables possibilités offertes par le format TEI. Dès lors, l'objectif d'exprimer une édition numérique scientifique (avec des critères nécessairement élaborés), dans un format partagé, repose sur l'adoption d'usages communs et, en fin de compte, la restriction des possibles de la TEI pour le type de document traité.

La ressource en ligne *Encoding Correspondence*² propose un ensemble de solutions d'encodage pour les écrits de correspondance. Ces solutions ont été adoptées dans le cadre du projet DAHN pour l'édition de la correspondance de Paul d'Estournelles de Constant (1852-1924)³. À la faveur de ce travail, F. Chiffolleau a produit un guide d'encodage fonctionnel (*guidelines*)⁴ au format TEI. C'est sur la base de cette documentation que l'encodage TEI pour la correspondance de C. de Salm a été élaboré.

Si l'encodage représente la forme finale que l'on souhaite donner à l'édition, y parvenir suppose un ensemble d'opérations qu'il s'agissait d'automatiser autant que possible, un certain nombre de tâches d'encodage final devant nécessairement être effectué manuel-

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

2. Stefan Dumont, Susanne Haaf et Sabine Seifert, *Encoding Correspondence. A Manual for Encoding Letters and Postcards in TEI-XML and DTABf*, 2019-2020, URL : <https://encoding-correspondence.bbaw.de/v1/index.html> (visité le 25/07/2022).

3. F. Chiffolleau, *Encoding an XML Tree Model for My Corpus*, Digital Intellectuals, 25 mars 2020, URL : <https://digitalintellectuals.hypotheses.org/3360> (visité le 25/07/2022).

4. Id., DAHN Project...

lement en bout de chaîne.

Cette partie détaille les phases de la transformation des transcriptions obtenues par HTR. Il a dans un premier temps fallu procéder à un choix technologique : celui du projet qui allait pouvoir servir de modèle.

3.1 Choisir un projet-modèle

Deux solutions concurrentes ont pu être envisagées grâce aux outils produits par d'autres entreprises éditoriales. Le projet DAHN avait donné naissance à un ensemble de scripts Python permettant de transformer le texte d'une lettre en encodage TEI, sélectionnant les lignes du texte selon les motifs qu'ils contiennent (patronymes, dates, signature, etc.) pour les inscrire dans les éléments correspondants du format TEI. Cette méthode avait le désavantage de ne pas permettre d'exploiter l'annotation des régions et des lignes d'écriture développée lors de la segmentation des lettres⁵.

Par bonheur, un stage simultané à celui-ci, mené par Kelly Christensen, s'était attelé au développement d'une application en langage Python, Alto2tei⁶, capable de transformer une reproduction numérique de document textuel publiée sur Gallica en intégrant dans l'encodage TEI les annotations des régions et des lignes d'écritures contenues dans les transcriptions au format ALTO. On pouvait dès lors envisager d'exploiter les annotations que l'on avait faites sans devoir reconstituer la structure sémantique du texte par une méthode secondaire.

Embarquant tout le contenu structuré des transcriptions-sources, cette application offre en outre l'avantage d'inscrire de manière pérenne le résultat des transcriptions dans l'édition électronique elle-même. Elle constitue ainsi l'archive sur laquelle se fonde l'encodage et rend également possible de générer de nouveaux fichiers ALTO pour les utiliser en tant que vérités de terrain⁷. Cette archive documente enfin le contexte de l'édition, puisque dans le cas où la ou les doubles pages attestant la lettre traitée attestent également d'autres lettres (avant ou après), c'est l'intégralité des transcriptions de ces documents qui est présentée dans cette partie du fichier TEI⁸.

5. Le projet avait été mené avant qu'eScriptorium ne propose une interface permettant cette annotation sémantique.

6. L'application que l'on a reprise est la suivante : K. Christensen, S. Gabay et A. Pinche, *Alto 2 Tei*, 6 mai 2022, URL : <https://github.com/kat-kel/alto2tei> (visité le 25/07/2022). Elle a elle-même été développée à partir d'une application développée par Vincent Jolivet, chef de la mission numérique de l'École nationale des chartes (ENC) : V. Jolivet, *Alto2tei*, École nationale des chartes, 12 mai 2022, URL : <https://github.com/chartes/alto2tei> (visité le 25/07/2022).

7. Cette archive est placée dans l'élément <sourceDoc> du fichier TEI.

8. Ceci n'est néanmoins valable que pour les textes inventoriés, car tenir compte également des textes non-inventoriés eut empêché l'automatisation de la construction des fichiers TEI.

3.2 Distribuer les fichiers de chaque lettre dans des dossiers

L'application Alto2tei de Gallic(orpor)a a été conçue pour produire un fichier TEI unique à partir d'un dossier contenant un ensemble de fichiers ALTO. Il s'agissait donc dans un premier temps de distribuer, à partir d'un lot de fichiers ALTO contenant les transcriptions d'une succession de lettres, de distribuer dans des dossiers distincts, les seuls fichiers pertinents pour chaque lettre, car, encore une fois, l'objectif était d'éditer chaque lettre dans un fichier TEI propre.

Les informations nécessaires à la distribution des fichiers pouvaient être récupérées dans un fichier de données produit à une étape antérieure du travail, lorsqu'il s'agissait de sélectionner les seules images contenant des lettres inventoriées⁹. L'extrait suivant montre parmi les notices (*records*) relatives à un dossier, que la lettre ayant pour identifiant CdS-b1-06pa est attestée sur deux fichiers-images. Ainsi, le script que l'on a élaboré pour la distribution des fichiers¹⁰ crée un dossier portant le nom de l'identifiant de la lettre et place dans ce dossier les images ainsi que les fichiers ALTO correspondants.

```
{
  "records": {
    "CdS-b1-06pa": {
      "incipit": "Madame, je me suis empressé...",
      "URL": "https://constance-de-salm.de/archiv/#/document/8886",
      "init_image": "CdS02_Konv002-03_0056.jpg",
      "title_position": 3,
      "images": [
        "CdS02_Konv002-03_0056.jpg",
        "CdS02_Konv002-03_0057.jpg"
      ]
    }
  }
}
```

Une fois les fichiers distribués, l'application Alto2tei est capable de générer un fichier TEI pour chaque lettre. Mais l'étape suivante représente davantage de difficulté. Une fois le contenu des différents fichiers ALTO intégré au fichier TEI de chaque lettre, il faut que la construction de l'élément `<text>` du fichier TEI s'opère de manière sélective, en extrayant de son contexte la lettre à éditer sans les textes des lettres voisines, comme l'explique le point suivant.

9. Voir *supra*, p. 27.

10. S. Biay, *distributionFichiers.py*, 4 juil. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/py/distributionFichiers.py>.

3.3 Récupérer les zones d'écritures pertinentes

L'exemple de la lettre CdS-b1-06pa permet d'illustrer les difficultés de cette étape. L'image 3.1 présente la double page attestant le début de la lettre (en bas à droite), tandis que l'image 3.2 en présente la fin (en haut à gauche).

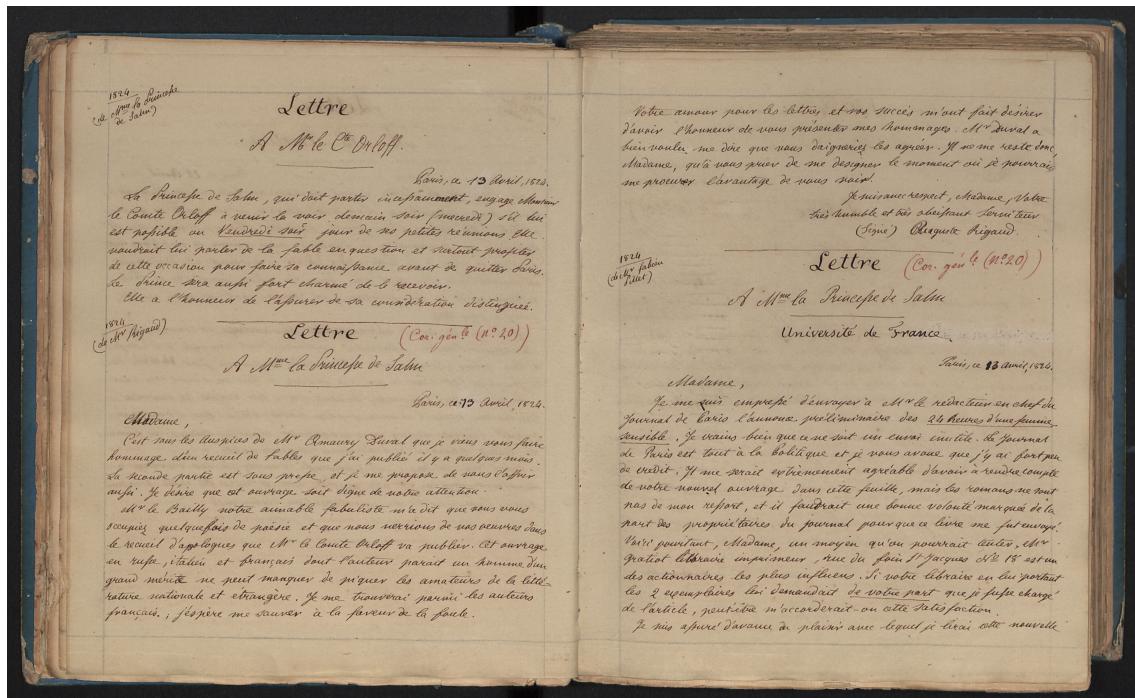


FIGURE 3.1 – Page présentant deux lettres puis le début de la lettre CdS-b1-06pa (notice *CdS/02_3/056-057*, URL : <https://constance-de-salm.de/archiv/#/document/8886> (visité le 21/06/2022)).

Dans un premier temps, l'application Alto2tei importe dans l'élément <sourceDoc> du fichier TEI de la lettre l'intégralité du contenu des deux fichiers ALTO, y compris le texte des autres lettres attestées sur les mêmes pages. Pour sélectionner le seul texte de la lettre à éditer, on s'est appuyé sur l'existence d'un titre pour chaque lettre. Ce titre a fait l'objet d'une annotation lors de la phase de segmentation et ses lignes sont donc accompagnées de l'information sémantique **HeadingLine :title**.

Un script Python a été conçu pour intervenir dans l'application Gallic(orpor) a entre la construction du <sourceDoc> et la construction du <text> pour sélectionner les lignes pertinentes à éditer¹¹. Ce script doit récupérer dans un premier temps une information capitale : la position, dans la double page, du titre de la lettre à éditer. Connaissant le nombre de lettres attestées sur une même page et sachant que leurs identifiants se succèdent dans l'ordre alphanumérique, il avait été possible de calculer cette position automatiquement en générant le fichier de données dont on a montré un extrait précédemment (l'informa-

11. Il s'agit de la fonction `selectionBlocs` dans le script Id., *cdsFonctions.Py*, 22 juil. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/4ea56b3694faaba079fa7a26a9032d6a0a99513b/transformation-alto-tei/py/cdsFonctions.py> (visité le 27/07/2022).

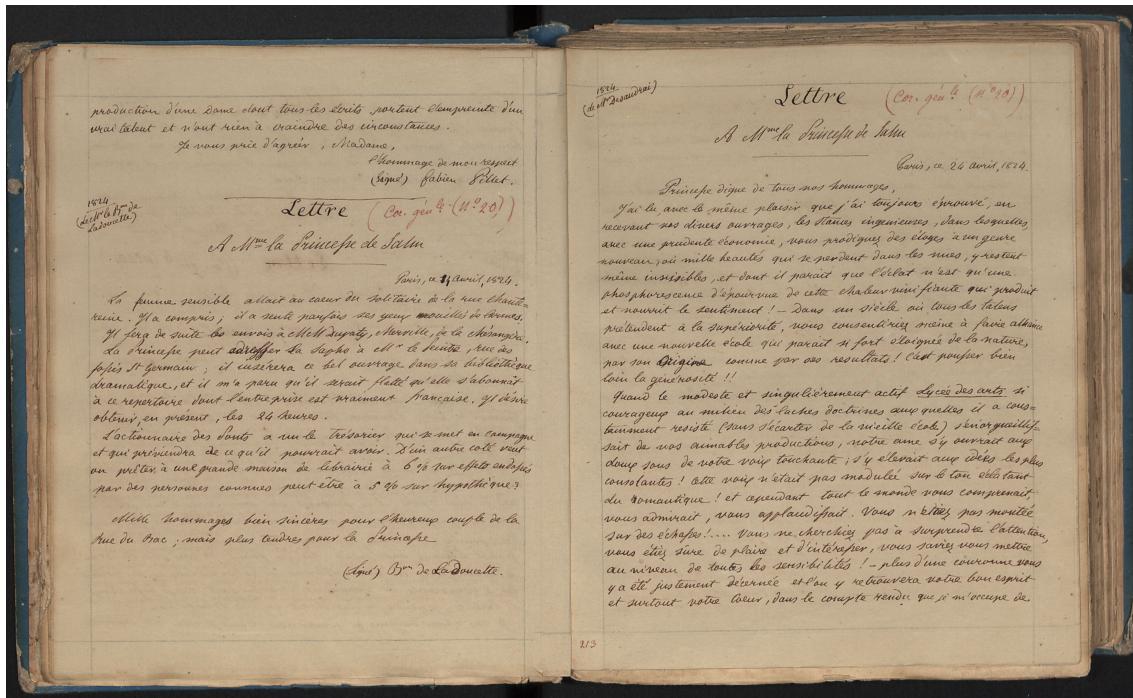


FIGURE 3.2 – Page présentant la fin de la lettre CdS-b1-06pa puis deux autres lettres.

tion est donnée sous la clé `title_position`)¹². On peut vérifier l'information en se reportant à l'image 3.1 : le titre de la lettre d'exemple, dont le texte commence par *Madame, je me suis empressé* est bien le troisième titre de la page. Le titre de la lettre ayant été localisé par sa position parmi les régions d'écriture, on procède alors à la récupération des régions d'écritures qui le suivent sur cette double page et sur la suivante, jusqu'à recontrer une nouvelle région d'écriture contenant un titre, ce qui indique qu'une nouvelle lettre est en train de commencer et que la sélection des régions pertinentes doit donc être interrompue¹³. Ces régions d'écritures récupérées, on en extrait la liste des identifiants de chaque ligne, car le format ALTO attribue à chaque ligne un identifiant unique. C'est cette liste d'identifiant qui a permis de poser un filtre lors de la construction de l'élément `<text>` du fichier TEI à partir du contenu de son élément `<sourceDoc>` : l'application Alto2tei a été modifiée pour ne construire l'élément `<text>` qu'à partir des lignes dont l'identifiant a été listé comme pertinent, les autres lignes étant simplement ignorées.

L'extrait de données suivant montre l'information structurée (correspondant à une seule ligne de texte) à partir de laquelle est effectuée la construction du `<text>` :

[

```
"CdS02_Konv019_0003_z1_l16",
"16",
"recommandé à tous a part de ne point inquiéter...",
```

12. Voir *supra*, p. 65

13. Cette opération de sélection a été effectuée au moyen de requêtes X-path construire dans le script Python grâce à la librairie Lxml (<https://lxml.de/>).

```

"DefaultLine",
>MainZone",
"CdS02_Konv019_0003_z1",
"CdS02_Konv019_0003",
>eSc_line_f4af3b05"
]

```

On constate que le texte de la ligne est transmis au script de construction accompagné de nombreuses métadonnées, dont l'identifiant unique de la ligne issu du fichier ALTO (dernière donnée de la liste : ici `eSc_line_f4af3b05`), ainsi que deux informations essentielles pour la suite des opérations : le type de région d'écriture auquel la ligne appartient (ici `MainZone`) et le type de ligne lui-même (ici `DefaultLine`). Ces informations permettent en effet de structurer l'encodage de la lettre, comme on doit l'expliquer à présent.

3.4 Structurer l'encodage d'une lettre

L'opération la plus délicate de la transformation TEI est la construction de l'élément `<text>` à partir des lignes filtrées comme pertinentes. Le projet Gallic(orpor)a ayant vocation à transformer en TEI une très large typologie de documents, il propose une structuration très générique des textes édités. En revanche, une édition de correspondance concerne une typologie de document restreinte et implique une structuration d'encodage particulière. Ainsi le texte des lettres doit être sémantiquement structuré en trois parties principales :

1. Un *opener* contenant tous les éléments graphiques disposés visuellement au début de la lettre (en-tête, titre, date, nom de l'expéditeur, salutation) ;
2. Une succession de paragraphes ;
3. Un *closer* contenant les éléments graphiques disposés visuellement en fin de lettre (salutation, signature) et qui peut éventuellement être suivi d'un post-scriptum.

Le fait que le contenu de chaque ligne soit transmis au script de construction accompagné de ses métadonnées (et notamment des types de régions et de lignes associés au contenu) a permis d'automatiser cette structuration. Mais il fallait pour cela réécrire de manière profonde le script sur lequel l'application Alto2tei de Gallic(orpor)a reposait¹⁴. Cette réécriture exprime un ensemble de conditions relativement complexe, mais dont le principe général est simple : ni le *opener* ni le *closer* de la lettre ne contiennent de ligne dont le type soit `DefaultLine`. Dès lors, l'énumération de chaque ligne dans l'ordre du texte permet de définir la partie dans laquelle on se situe :

14. Pour le script original, voir K. Christensen, `body_build.py`, 6 mai 2022, URL : https://github.com/kat-kel/alto2tei/blob/main/src/body_build.py (visité le 27/07/2022) ; pour son adaptation au présent proje, voir S. Biay, `build_body.py`, 22 juil. 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/py/build_body.py.

1. Tant que l'on ne rencontre pas de **DefaultLine**, on se situe dans le *opener* ;
2. Dès que l'on rencontre une **DefaultLine**, on se situe dans le corps de la lettre ;
3. Dès que l'on rencontre une mention de date, une salutation ou une signature, on se situe dans le *closer* (celui-ci pouvant être éventuellement suivi par un post-scriptum).

L'arbre général des conditions de traitement des lignes est représenté par la figure 3.3.

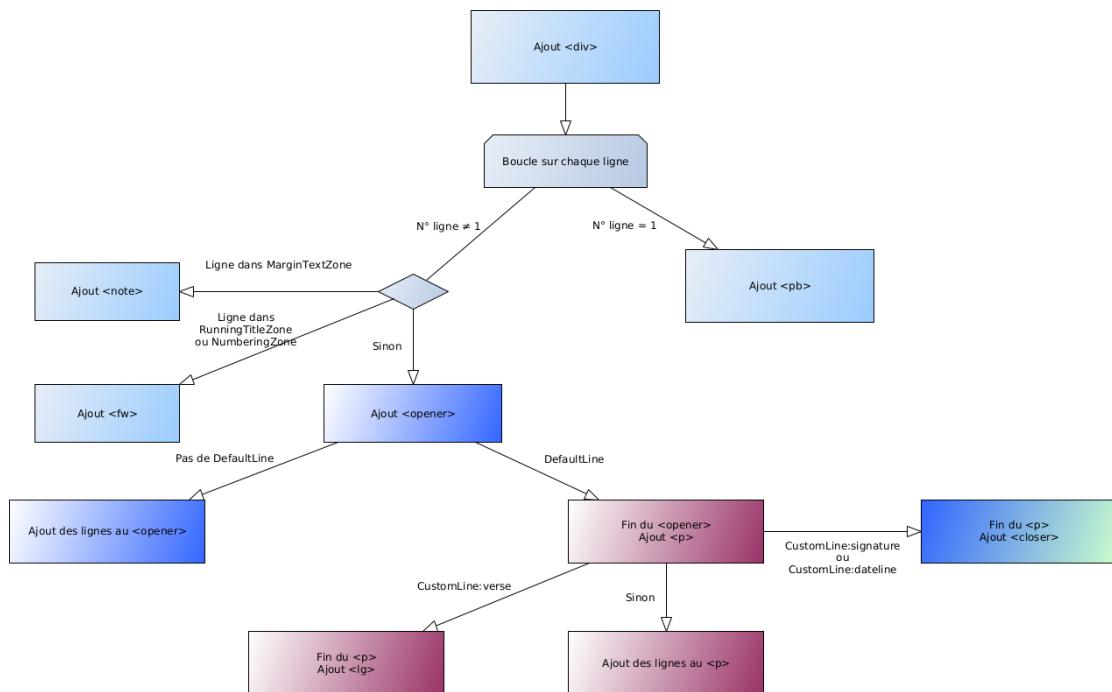


FIGURE 3.3 – Arbre des conditions de traitements des lignes pour la construction de l'édition (image en haute définition disponible ici).

Outre les principales parties du texte, il fallait tenir compte des éléments marginaux : systèmes de numérotation (page, lettre), notes, annotations érudites ou d'origine, titres courants. Il n'était pas possible d'associer strictement ces éléments à leur lettre. Lorsque plusieurs pages se partagent la même lettre, la numérotation des éléments marginaux ne s'inscrit pas toujours au bon endroit dans la succession des régions d'écriture (ces éléments sont souvent rejetés en fin de numérotation). Pour que ce problème ne passe pas inaperçu, l'encodage automatique attire l'attention de l'éditeur de la façon suivante :

```

<note type="MarginTextZone">
  <!--Vérifier que la note corresponde à la lettre et mettre à jour le type-->
  [transcription de la note]
</note>
  
```

Une dimension importante de la complexité de l'encodage des lettres a été l'alternance présentée par certaines entre contenu prosaïque (à transformer en éléments <p>) et contenu versifié (à transformer en élément lg). Encore une fois, l'annotation des lignes permet de les traiter de la manière adéquate (CustomLine :verse dans le cas des parties versifiées).

On présente ici à titre d'exemple le résultat de l'encodage automatique d'un texte singulier dans sa constitution, car il contient un poème en tant que post-scriptum.

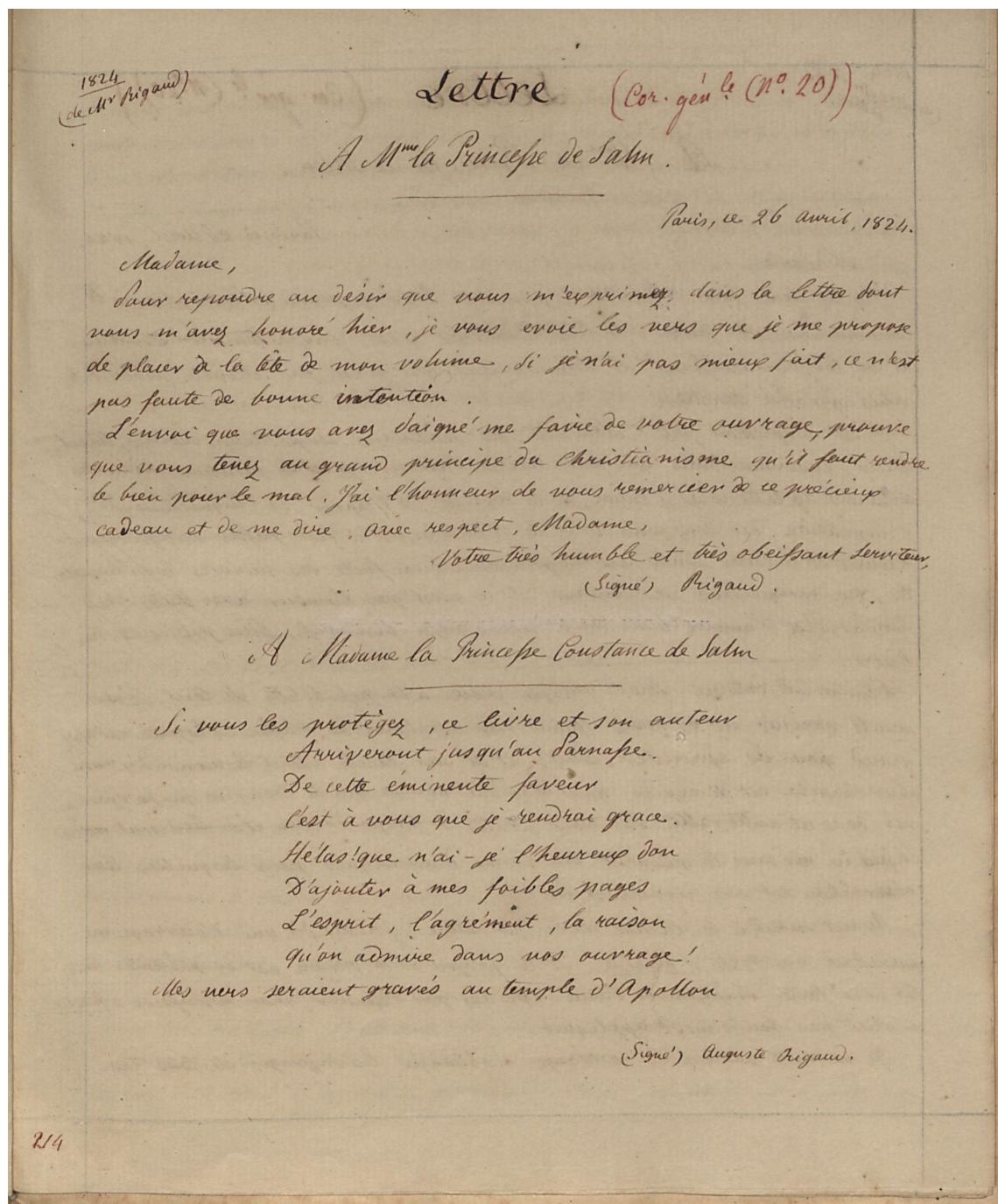


FIGURE 3.4 – Lettre de P.-A. Rigaud à C. de Salm (notice CdS/02_3/059, URL : <https://constance-de-salm.de/archiv/#/document/8889>)).

```
<div>
  <opener>
    <fw type="letterhead"><lb/>1824<lb/>(de Mr Rigaud)</fw>
    <title><lb/>Lettre<lb/>A Mme la Princesse de Salm.</title>
    <note><lb/>(Cor. génle. (n° 20))</note>
    <dateline><lb/>Paris, ce 26 avril, 1824.</dateline>
    <salute><lb/>Madame,</salute>
  </opener>
  <p><lb/>vous m'avez honoré hier, je vous evoie les vers que je me propose
    <lb/>Pour repondre au désir que vous m'exprimez, dans la lettre dont<lb/>
    de placer de la tête de mon volume, Si je n'ai pas mieux fait, ce n'est
    <lb/>pas faute de bonne intention.<lb/>L'envoi que vous avez daigné me
    faire de votre ouvrage, prouve<lb/>que vous tenez au grand principe du
    Christianisme qu'il faut rendre<lb/>le bien pour le mal. J'ai l'honneur
    de vous remercier de ce précieux<lb/>cadeau et de me dire, avec respect,
    Madame,
  </p>
  <closer>
    <signed><lb/>Votre très humble et très obéissant Serviteur,
    <lb/>(Signé) Rigaud.</signed>
  </closer>
  <postscript>
    <p><lb/>A Madame la Princesse Constance de Salm</p>
    <lg>
      <l><lb/>Si vous les protégez, ce livre et son auteur</l>
      <l><lb/>Arriveront jusqu'au Parnasse.</l>
      <l><lb/>De cette éminente faveur</l>
      <l><lb/>C'est à vous que je rendrai grace.</l>
      <l><lb/>Hélas ! que n'ai-je l'heureux don</l>
      <l><lb/>D'ajouter à mes faibles pages</l>
      <l><lb/>L'esprit, l'agrément, la raison</l>
      <l><lb/>Qu'on admire dans vos ouvrage !</l>
      <l><lb/>Mes vers seraient gravés au temple d'Apollon</l>
    </lg>
    <signed><lb/>(Signé) Auguste Rigaud.<!--Salut--></signed>
  </postscript>
  <fw corresp="#CdS02_Konv002-03_0059_z4" type="NumberingZone">
    <lb/>214<!--Vérifier que le numéro corresponde à la lettre et
    mettre à jour le type : pageNum ou letterNum -->
  </fw>
</div>
```

Enfin, les corrections écrites dans l’interligne ont elles aussi été annotées de façon particulière (`InterlinearLine`), ce qui permet de les traiter dans un encodage qui exprime leur fonction de correction du texte original.

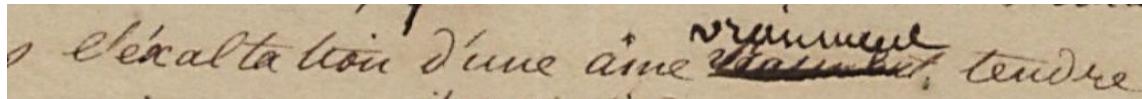


FIGURE 3.5 – Exemple de correction interlinéaire (notice `CdS/02_3/072`, URL : <https://constance-de-salm.de/archiv/#/document/8909>).

L’encodage automatique prend la forme suivante (les balises de début de ligne ont été supprimées pour davantage de clarté) :

```

l'exaltation d'une âme [.....] tendre et
<!--Correction interlinéaire-->
<choice>
  <sic/><!--Partie laissée vide car non prédictible par le script-->
  <corr>
    vraiment
  </corr>
</choice>
```

Bien entendu, cet encodage ne constitue qu’une étape intermédiaire, laissant une partie de travail non automatisable à la responsabilité de l’éditeur, qui en l’occurrence doit préciser que la partie corrigée n’est pas lisible :

```

l'exaltation d'une âme
<choice>
  <sic>
    <unclear reason="illegible"/>
  </sic>
  <corr>
    vraiment
  </corr>
</choice> tendre et
```

Le bon fonctionnement des scripts d’encodage a été testé sur une sélection de lettres, choisies pour la diversité des cas de figures qu’elles présentaient¹⁵. Il s’agissait de structurer correctement des lettres se développant sur plusieurs pages (sur deux pages ou sur plus de deux pages), mais aussi de bien distinguer plusieurs lettres écrites sur la même page,

¹⁵. La collection de test se trouve à l’adresse suivante : <https://github.com/sbiay/CdS-edition/tree/main/transformation-alto-tei/collection-test>.

cas particulièrement fréquent dans les recueils de copies. L'automatisation de l'encodage a été testée tout particulièrement avec ces recueils de copies, documents très structurés, comportant des manchettes, des notes infrapaginaires, des corrections, des textes en vers, et où les lettres se succèdent sans laisser de blanc. Il va de soi que la bonne structuration de l'encodage de ces documents complexes garantit la bonne structuration des documents plus simples.

Dans l'éventualité où une lettre serait dépourvue de titre, élément-clé de la méthode d'encodage que l'on a élaborée, la production de l'édition ne serait pas pour autant rendue impossible : le contenu du texte serait encodé malgré tout, avec le texte des documents qui le précèdent ou qui le suivent sur la même page, obligeant à la suppression manuelle de ces parties.

3.5 Incrire les métadonnées dans l'en-tête de la lettre

3.5.1 Décrire la lettre

Le projet Gallic(orpor)a ne traitant que des sources publiées sur Gallica, il récupère les métadonnées bibliographiques dans deux sources principales : les manifestes 3IF¹⁶ des documents et le catalogue général de la BnF.

Les métadonnées de la correspondance de C. de Salm ont quant à elle été publiées sur la plateforme Zenodo dans un format tabulaire¹⁷. Il est donc aisément d'associer le fichier TEI à ses métadonnées en allant chercher dans ces tableaux l'identifiant de la lettre à éditer. Le script `build_teihandler` de l'application Alto2tei a été modifié pour procéder à cette récupération inscrire des données choisies dans l'arborescence de l'en-tête du fichier TEI. L'inscription de ces données a lieu au niveau de l'élément `<profileDesc>`, au sein d'un sous-élément spécialement dévolu à la description des correspondances : le `<correspDesc>`. Celui-ci comporte des éléments obligatoires (`<correspAction>`) pour la description de l'expédition (expéditeur, lieu, date) et de la réception d'une lettre.

```
<profileDesc>
  <correspDesc>
    <correspAction type="sent">
      <persName ref="https://viaf.org/viaf/121051">
        Salm, Constance de (CdS)
      </persName>
```

16. Il s'agit de fichiers au format Json. Pour un bref aperçu des principes de ce mode d'exposition des données, voir Régis Robineau, *Comprendre IIIF et l'interopérabilité des bibliothèques numériques*, Insula, 8 nov. 2016, URL : <https://insula.univ-lille.fr/2016/11/08/comprendre-iiif-interoperabilite-bibliotheques-numeriques/> (visité le 27/07/2022).

17. Florence de Peyronnet-Dryden, Christiane Coester, Eva Dade, Eva Knels, Hannah Schneider, Sabine Breihof et Alice Habersack, *Inventar Der Korrespondenz Der Constance de Salm (1767-1845)*, 17 nov. 2021, URL : <https://zenodo.org/record/5707822> (visité le 27/07/2022).

```

<placeName ref="https://www.geonames.org/2894637">
    Dyck (Schloss), Gde. Jüchen
</placeName>
<date when-iso="1816-01-14">1816-01-14</date>
</correspAction>
<correspAction type="received">
    <persName ref="n/a">Prous, Henry</persName>
</correspAction>
</correspDesc>
</profileDesc>

```

Il convient également de donner une expression plus littéraire à ces données. C'est le rôle de l'élément titre (`<title>`) du document TEI que l'on a forgé de manière automatique, par la manipulation des chaînes de caractères du nom de l'expéditeur, du destinataire, du lieu et de la date d'expédition (avec les mentions s.l. et s.d. dans le cas de lettres sans lieu ou sans date). Il s'agissait notamment de simplifier les noms (débarrassés des titres de noblesse) et de rétablir le prénom avant le patronyme, ou encore d'exprimer la date en toutes lettres. Toujours pour la même lettre (et donc à partir des mêmes données brutes), le titre a été exprimé de la façon suivante :

```

<title>
    Lettre de Constance de Salm à Henry Prous (Dyck, le 14 janvier 1816)
</title>

```

3.5.2 Citer la notice de l'inventaire

Il était également essentiel d'inscrire dans l'en-tête du fichier TEI la référence bibliographique de la notice de la lettre publiée en ligne. Cette description intervient dans l'élément `<sourceDesc>` :

```

<sourceDesc>
    <biblStruct>
        <analytic>
            <title>CdS/19/002</title>
            <idno>https://constance-de-salm.de/archiv/#/document/10462</idno>
        </analytic>
        <monogr>
            <title>
                La correspondance de Constance de Salm (1767-1845).
                Inventaire du fonds Salm de la Société des Amis du Vieux Toulon et de sa Région
                et du fonds Constance de Salm, Archiv Schloss Dyck (Mitgliedsarchiv
                der Vereinigten Adelsarchive im Rheinland e.V.). Édition numérique
            </title>

```

```

</title>
<imprint>
    <publisher>DHI Paris</publisher>
    <pubPlace>Paris</pubPlace>
    <date>2021</date>
</imprint>
</monogr>
</biblStruct>
</sourceDesc>
```

3.5.3 Renseigner les données du projet

Enfin, les données descriptives du projet, l'édition de la correspondance dans son ensemble, ont été décrites dans l'élément `<seriesStmt>` de la façon suivante (l'encodage des noms a été simplifié) :

```

<seriesStmt>
    <title>La correspondance de Constance de Salm (1767-1845)</title>
    <respStmt>
        <resp>Encadrement scientifique et technique du projet</resp>
        <persName>Anne Baillot</persName>
        <persName>Mareike König</persName>
        <persName>Floriane Chiffoleau</persName>
    </respStmt>
    <respStmt>
        <resp>Réalisation de la chaîne de traitement</resp>
        <persName>Sébastien Biay</persName>
    </respStmt>
</seriesStmt>
```

3.6 Finaliser l'encodage d'une lettre

Un certain nombre de tâches non automatisables ont fait l'objet d'une documentation fonctionnelle, exemples à l'appui¹⁸.

La première consiste à diviser les paragraphes. En effet, les zones d'écriture définies au moment de la segmentation englobe les paragraphes d'une manière indistincte, et seul l'oeil humain permet de resconstituer cette structure à posteriori.

Les corrections interlinéaires, en partie encodées automatiquement, doivent être réinsérées dans la ligne où se trouve la partie corrigée. On l'a également dit plus haut : les

^{18.} S. Biay, *Finaliser l'encodage d'une lettre*, 26 juil. 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/Finaliser_encodage.ipynb (visité le 26/07/2022).

notes marginales et éléments de numérotation doivent être contrôlées et parfois déplacées.

Il n'a pas été prévu de script pour la résolution automatique des abréviations courantes, ce qui aurait été possible avec davantage de temps bien entendu.

D'une manière générale, l'édition numérique de la correspondance de C. de Salm, avec les subtilités d'encodage inhérentes à ce type de travail, comme la création d'un appareil critique pour corriger le texte là où le copiste a commis des fautes manifestes, ne peut pas se satisfaire d'une simple documentation fonctionnelle. L'enjeu appelle la rédaction d'un schéma ODD (*One Document Does it all*) qui établisse de manière précise et exhaustive les règles d'encodage de la correspondance et en porte la spécification technique¹⁹. Sa mise en œuvre étant un travail exigeant, les dimensions du stage ne permettaient pas d'atteindre cet objectif supplémentaire²⁰.

19. Une ODD joue à la fois un rôle de documentation pour les contributeurs du projet, sorte de guide d'encodage, et de validation technique des documents.

20. On s'est contenté de finaliser manuellement l'encodage de trois petites lettres, afin de pouvoir en évaluer les possibilités d'exploitation. Cf. <https://github.com/sbiay/CdS-edition/tree/main/transformation-alto-tei/final>

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).

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'ils 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

prédition Transcription automatique de lignes d'écriture par un algorithme de reconnaissance des caractères.. 2, 16, 18, 22, 36, 50–58

segmentation Analyse optique d'une image permettant d'obtenir la reconnaissance des régions et des lignes d'écriture.. 2, 7–9, 42, 48, 49, 75

Acronymes

- ALTO** *Analyzed Layout and Text Object* (format XML pour la description des textes et de leur mise en page). 8, 36, 41, 50, 53, 54, 56, 64–68
- ATILF** Analyse et Traitement Informatique de la Langue Française (Centre National de la Recherche Scientifique-Université de Lorraine). 58
- BnF** Bibliothèque nationale de France. 8, 28, 73
- C. de Salm** Constance de Salm. 5, 21–23, 25, 30, 43, 47, 51, 52, 55, 58–60, 63, 70, 73, 83
- Cremma** Consortium Reconnaissance d’Écriture Manuscrite des Matériaux Anciens. 9, 11, 12, 15, 16, 30, 47, 48, 84
- DAHN** Dispositif de soutien à l’Archivistique et aux Humanités Numériques (partenariat entre l’Inria (équipe ALMAnaCh), l’Université du Mans et l’EHESS). 5, 63, 64
- DHIP** Deutsches Historisches Institut Paris. 5, 30, 43
- ENC** École nationale des chartes. 64
- FuD** Die Virtuelle Forschungsumgebung für die Geistes- und Sozialwissenschaften. 5
- Gallic(orpor)a** Gallic(orpor)a (projet financé par Huma-Num et le BnF DataLab). 18, 65, 66, 68, 73
- HTR** *Handwritten Text Recognition*. 7, 11–16, 18, 21, 30, 47, 55, 64
- Json** *JavaScript Object Notation* (format standard de représentation de données structurées). 25, 27, 56, 59, 60, 73
- LAI** Lettre Absente de l’Inventaire en ligne. 23, 24, 31, 34, 35
- Lectaurep** Lecture Automatique de Répertoires. 21, 22, 43, 50–52
- OCR** *Optical Character Recognition*. 7, 30

SegmOnto SegmOnto : A Controlled Vocabulary to Describe the Layout of Pages.
17, 33, 38–40

TEI Text Encoding Initiative. 5, 32, 40, 55, 58, 63–68, 73, 74

XML Extensible Markup Language. 63

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_3/056-057, URL : <https://constance-de-salm.de/archiv/#/document/8886> (visité le 21/06/2022).

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_3/059, URL : <https://constance-de-salm.de/archiv/#/document/8889>.

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

CdS/02_3/072, URL : <https://constance-de-salm.de/archiv/#/document/8909>.

- CdS/19/036-037*, URL : <https://constance-de-salm.de/archiv/#/document/10504>.
- CdS/19/054-056*, URL : <https://constance-de-salm.de/archiv/#/document/10517>
(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), *build_body.py*, 22 juil. 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/py/build_body.py.
- *dictGenerateur.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/dictGenerateur.py>.
 - *distributionFichiers.py*, 4 juil. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/py/distributionFichiers.py>.
 - *donneesImages.py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesImages.py>.
 - *donneesNonPubliees.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/donneesNonPubliees.py>.
 - *Finaliser l'encodage d'une lettre*, 26 juil. 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/transformation-alto-tei/Finaliser_encodage.ipynb (visité le 26/07/2022).
 - *injectTranscript.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/injectTranscript.py>.
 - *journalReconn.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/journalReconn.py>.
 - *Mains*, Éditer la correspondance de Constance de Salm (1767-1845), 10 juin 2022, URL : <https://github.com/sbiay/CdS-edition/tree/main/htr/mains>.
 - *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.
 - *supprLignesVides.py*, 20 mai 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/supprLignesVides.py>.

- *Tester et entraîner un modèle de reconnaissance d’écriture*, 20 mai 2022, URL : https://github.com/sbiay/CdS-edition/blob/main/htr/Tester_et_entraîner_un_modèle_HTR_avec_Kraken.ipynb.
- BIAY (Sébastien) et CHIFFOLEAU (Floriane), *spellcheckTexts.py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/spellcheckTexts.py>.
- *textCorrection.py*, 6 avr. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/main/htr/py/textCorrection.py>.

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).

PEYRONNET-DRYDEN (Florence de), COESTER (Christiane), DADE (Eva), KNELS (Eva), SCHNEIDER (Hannah), BREIHOFER (Sabine) et HABERSACK (Alice), *Inventar Der Korrespondenz Der Constance de Salm (1767-1845)*, 17 nov. 2021, URL : <https://zenodo.org/record/5707822> (visité le 27/07/2022).

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), *Encoding an XML Tree Model for My Corpus*, Digital Intellectuals, 25 mars 2020, URL : <https://digitalintellectuals.hypotheses.org/3360> (visité le 25/07/2022).

- *[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).

- CHIFFOLEAU (Floriane), *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).
- CHRISTENSEN (Kelly), *body_build.py*, 6 mai 2022, URL : https://github.com/kat-kel/alto2tei/blob/main/src/body_build.py (visité le 27/07/2022).
- CHRISTENSEN (Kelly), GABAY (Simon) et PINCHE (Ariane), *Alto 2 Tei*, 6 mai 2022, URL : <https://github.com/kat-kel/alto2tei> (visité le 25/07/2022).
- 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).
- DUMONT (Stefan), HAAF (Susanne) et SEIFERT (Sabine), *Encoding Correspondence. A Manual for Encoding Letters and Postcards in TEI-XML and DTABf*, 2019-2020, URL : <https://encoding-correspondence.bbaw.de/v1/index.html> (visité le 25/07/2022).
- Frantext*, URL : <https://www.frantext.fr/> (visité le 22/06/2022).
- JOLIVET (Vincent), *Alto2tei*, École nationale des chartes, 12 mai 2022, URL : <https://github.com/chartes/alto2tei> (visité le 25/07/2022).
- 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

- BAUDRY (Hervé), « Les archives inquisitoriales (Portugal) sous HTR : le projet TraPrInq (Transcribing the court records of the Portuguese Inquisition, 1536-1821) », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- BIAY (Sébastien), *cdsFonctions.Py*, 22 juil. 2022, URL : <https://github.com/sbiay/CdS-edition/blob/4ea56b3694faaba079fa7a26a9032d6a0a99513b/transformation-alto-tei/py/cdsFonctions.py> (visité le 27/07/2022).
- BIZAIS-LILLIG (Marie) et VIDAL-GORÈNE (Chahan), « Expérimentations pour l'analyse automatique de sources chinoises anciennes », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

- BOSCHETTI (Federico) et TOMMASI (Tatiana), « EpiSearch. Recognising Ancient Inscriptions in Epigraphic Manuscripts », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- CAMPS (Jean-Baptiste) et PINCHE (Ariane), « CremmaLab Projects : Transcription Guidelines and HTR Models for French Medieval Manuscripts », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- 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).
- *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) et CLÉRICE (Thibault), « Sharing HTR Datasets with Standardized Metadata : The HTR-United Initiative », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 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).
- CONSTUM (Thomas), « Reconnaissance et extraction d'informations dans des tableaux manuscrits historiques : vers une compréhension des recensements de Paris de l'entre-deux guerre », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- CUÉLLAR (Álvaro), « Un modèle ouvert pour la reconnaissance automatique des manuscrits du théâtre espagnol du Siècle d'Or », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- Documents anciens et reconnaissance automatique des écritures manuscrites (HTR)*, colloque, 23 et 24 juin 2022, École nationale des chartes, Paris, programme et résumés, URL : <https://cremmalab.hypotheses.org/colloque-htr-programme> (visité le 03/05/2022).
- FIZAINE (Florian) et BOUYÉ (Édouard), « Lettres en lumières », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 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).
- GABAY (Simon) et KÜNZLI (Pierre), « FONDUE - A Lightweight HTR Infrastructure for Geneva », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

- GABAY (Simon), PINCHE (Ariane) et CHRISTENSEN (Kelly), « SegmOnto – A Controlled Vocabulary to Describe Historical Textual Sources », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 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).
- KIESSLING (Benjamin) et STOKES (Peter A.), « New Developments in Kraken and eScriotorium », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- LEBLANC (Élina) et JACSONT (Pauline), « De Transkribus à eScriotorium : retour(s) d'expérience sur l'usage d'outils d'HTR appliqués à un corpus d'imprimés espagnols du XIXe siècle », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- « Leiden Conventions », dans *Wikipedia*, 2021, URL : https://en.wikipedia.org/w/index.php?title=Leiden_Conventions&oldid=1004624327 (visité le 05/05/2022).
- MARGUIN-HAMON (Elsa), « Discours d'ouverture et présentation des projets CREMMA et CREMMALAB », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- PARASKEVI (Platanou), « HTR of Handwritten Paleographic Greek Text as a Function of Chronology », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.
- PAUPE (Elodie), « Une cursive du XVIIe siècle », dans *Documents Anciens et HTR*, 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).

ROBINEAU (Régis), *Comprendre IIIF et l'interopérabilité des bibliothèques numériques*, Insula, 8 nov. 2016, URL : <https://insula.univ-lille.fr/2016/11/08/comprendre-iiif-interoperabilite-bibliotheques-numeriques/> (visité le 27/07/2022).

SCHWEYER (Anne-Valérie), BURIE (Jean-Christophe) et TIEN NAM NGUYEN, « Analyse, reconnaissance et indexation des manuscrits cham », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 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).

STÖKL BEN EZRA (Daniel), HAYIM (Lapin) et JABLONSKI (Pavel), « From HTR to Critical Edition : A Semi-Automatic Pipeline », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

STÖKL BEN EZRA (Daniel), RUSTOW (Marina) et WITTY (Deborah), « Segmentation Mode for Archival Documents with Highly Complex Layout », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 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.

TORRES AGUILAR (Sergio) et JOLIVET (Vincent), « Modélisation et affinage HTR pour les ms méd. : stratégies et évaluation », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.

TUFFÉRY (Christophe), « Retour d'expériences sur l'utilisation comparée de plusieurs dispositifs de transcription numérique d'archives de fouilles archéologiques », dans *Documents Anciens et HTR*, Paris, École nationale des chartes, 2022.