Contenu de la documentation

Présentation

Contexte

glscds (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 Sozialwissenschften (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 XML-TEI. On s'appuiera pour cela sur les instruments et la documentation produits dans le cadre du projet Digital Edition of historical manuscripts (DAHN), fondé sur l'édition de la correspondance de Paul d'Estournelles de Constant (1852-1924) [1].

Il s'agit en particulier d'identifier les points de difficultés 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 Chiffoleau, *DAHN Project*, GitHub, URL : `https://github.com/FloChiff/DAHNProject` (visité le 05/04/2022).

Chapitre 1

Reconnaissance automatique des écritures manuscrites

(*Handwritten Text Recognition* (HTR))

Comme pour le traitement des écritures imprimées, la reconnaissance automatique des écritures manuscrites recouvre deux phases indissociables et complémentaires :

1. 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 repérées et numérotées dans l'ordre de lecture (ce sans quoi la transcription produite serait inexploitable !) ;
2. La reconnaissance des écritures proprement dite, qui procède à l'identification de chaque caractère sur les lignes précédemment repérées.

Avant de mettre en place une méthode de travail, il est primordial d'évaluer les caractéristiques paléographiques des sources d'une part, et de définir les finalités du travail d'autre part. En outre, on discutera dans ce chapitre du choix de l'application à utiliser pour y procéder, et l'on justifiera le choix que nous avons fait quant aux sources sur lesquelles nous avons travaillé, l'intégralité de la correspondance n'ayant évidemment pu être traitée en quatre mois de stage.

1.1 Enjeux et tâches préliminaires

1.1.1 Des sources écrites par plusieurs mains

Quatre à cinq mains différentes ont été repérées jusqu'à présent dans la correspondance de glscds (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 mixte [2] 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 Salm [3] ;
2. Reprendre un modèle générique entraîné dans le cadre du projet Lectaurep pour en évaluer les performances.

1.1.2 Objectif : éditer le texte des lettres

Il est nécessaire d'aborder cette phase de la reconnaissance automatique d'écriture en pleine conscience de l'objectif à atteindre : en l'occurrence, l'édition du texte 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 question.

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

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.

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 prédiction d'une acuité voisine de 90% (insuffisante pour l'édition) et une prédiction d'une acuité supérieure à 95% [4].

1.1.3 Choisir des collections d'évaluation

Afin de donner les meilleurs chances aux tests à effectuer avec le modèle 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 six [5].

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 retenu pour les tests.

Nous avons également analysé les écritures du recueil de la correspondance adressée par J.P.E. Martini à C. de Salm afin d'élargir la variété de notre corpus de tests. Nous y avons distingué deux mains [6].

On a privilégié pour les corpus de test et d'entraîner des modèles des reproductions favorables à une bonne reconnaissance de l'écriture, évitant en particulier les problèmes de transparence qui font ressortir au recto l'encre du verso.

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 pour une écriture particulièrement cursive (compter environ deux semaines pour disposer d'une bonne vingtaine de pages).

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

5. Constance de Salm, *Correspondance générale, seconde copie, 1^er^ volume, 1785-1814*, URL : `https://constance-de-salm.de/archiv/#/document/11215` (visité le 11/04/2022) ; Id., *Correspondance générale, seconde copie, 2^e^ volume, 1815-1821*, URL : `https://constance-de-salm.de/archiv/#/document/11216` (visité le 11/04/2022) ; Id., *Correspondance générale, seconde copie, 3^e^ volume, 1822-1828*, URL : `https://constance-de-salm.de/archiv/#/document/11217` (visité le 12/04/2022).

6. Une présentation des mains peut être parcourue sur le dépôt du projet

1.1.4 Préparer le traitement d'un dossier

L'archive photographique de la correspondance de C. de Salm comporte des documents non inventoriés. Afin de n'engager dans notre chaîne de traitement que des documents effectivement inventoriés, nous avons consacré un *notebook* à la préparation du traitement d'un dossier [7].

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 associées à une notice de l'inventaire. Nous avons pour cela écrit un script python [8] 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 notice [9].

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.

1.1.5 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 applications [10]. 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.

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

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

9. Le fichier donne par ailleurs la liste des images qui ne sont liées à aucune notice de l'inventaire, ainsi qu'une présentation des mêmes données d'association image-notice, mais cette fois par image et non par notice, et ce afin de permettre le contrôle visuel des zones de texte à transcrire (cf. *infra*, 1.2.5, p. 13)

10. Nous avons assisté le 9 mai 2022 à l'atelier organisé au sein du Data-Lab de la Bibliothèque nationale de France (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).

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 avancé, 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 Kraken [11]), promeut les principes de la science ouverte de multiples manières (développement *open-source*, respects de standards des formats numériques, ouverture des 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 consitution de données d'entraînement. Il en découle naturellement que ces données déterminent, conditionnent les résultats obtenus par les modèles entraînés (quelles images ont été choisies, quels textes ont été transcrits pour parvenir à tel résultat). Pour comprendre le fonctionnement de ces modèles et leurs performances, il faut donc disposer d'une archive des données d'entraînement ; celles-ci doivent être exposées de manière transparente, et ainsi pouvoir être critiquées, analysées ou réutilisées. Ainsi, le logiciel Kraken permet (techniquement) et la communauté eScriptorium encourage (politiquement) la publication et le partage des vérités de terrain (qui

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

sont les véritables données brutes d'entraînement) ainsi que des modèles eux-mêmes [12].

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 incompatible avec l'utilisation de solutions logicielles clé en main. Les besoins particuliers de la recherche sont ainsi beaucoup mieux servis par l'emploi de briques logicielles indépendantes, modulables, entre lesquelles peuvent s'échanger les données dans des standards bien établis, plutôt que par le recours à des suites logicielles performantes mais aux fonctionnalités déterminées par une communauté de développement extérieure au projet. Le risque est en effet immense de devoir reconsidérer les attendus du projets à la découverte soudaine d'une fonctionnalité manquante ou plus souvent encore de l'impossibilité de personnaliser un mode d'expression des données [13].

Pour l'ensemble de ces raisons, nous avons opté pour l'utilisation d'eScriptorium et de Kraken dans le cadre de ce stage. Le flux de travail pourra sembler complexe à un utilisateur peu aguerri en matière d'ingénierie, mais en contrepartie une documentation fonctionnelle pas à pas a été rédigée grâce à la technologie du *Jupyter notebook* qui permet en toute théorie de mener l'intégralité des tâches que l'on a experimentées avec une expertise réduite.

Dans ce genre de configuration, une assistance pourra être requise pour l'étape la plus délicate en termes d'ingénierie : l'installation des applications nécessaires à la conduite du projet, celle d'eScriptorium étant le point le plus critique et l'installation des applications en langage Python pouvant également poser quelques difficultés. Nous avons en effet utilisé eScriptorium à partir d'une installation locale [14], faisant appel aux seules ressources d'un ordinateur portable, à savoir sans serveur ni carte graphique externe [15]. Cette méthode nous a permis de procéder à des entraînements de modèle à partir de petits volumes de vérités de terrain. Si des entraînements plus massifs s'avéraient nécessaires, il serait alors impératif de se tourner vers une infrastructure dotée de plus grandes capacités de calcul, ce que, par exemple, un partenariat entre le DHIP et le projet Consortium Reconnaissance d'Écriture Manuscrite des Matériaux Anciens (Cremma) rendrait possible.

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

13. 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).

14. 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).

15. 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

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

L'annotation des régions et des lignes d'écritures répond à deux fonctions distinctes :

— Permettre l'entraînement d'un modèle de segmentation ;

— Transformer leur contenu afin de l'affecter à des éléments déterminés de l'arborescence XML-TEI qu'il faudra construire [16].

Cette réflexion sur les besoins de la transformation vers le format TEI a été nourrie par les *Guidelines* de l'édition de correspondance du projet DAHN [17]. Par ailleurs, F. Chiffoleau a formulé une ontologie pour les régions et lignes des écrits de correspondance en langue française pour le XXe siècle [18] dans le cadre du projet SegmOnto : A Controlled Vocabulary to Describe the Layout of Pages (SegmOnto) [19]. Afin de rendre notre propre typologie générique et de pouvoir exploiter l'outil de validation d'annotation HTRUC [20], nous avons repris les types SegmOnto en exploitant les types `CustomZone` et `CustomLine` lorsqu'il était nécessaire de les personnaliser.

1.2.1 Régions

L'entraînement d'un modèle de segmentation à reconnaître et annoter automatiquement des types de régions d'écritures est un travail complexe. La mise en page des lettres répond à des principes clairs pour l'oeil humain ; il présente en revanche d'importantes variations métriques dans l'espace de la page. Le meilleur exemple de ces variations se trouve dans les recueils : les lettres ont été transcrites les unes à la suite des autres ; un début de lettre peut dès lors se trouver à n'importe quelle hauteur de la page.

Ajouter un commentaire sur l'espacement des lignes de l'en-tête

La reconnaissance de la fin d'une lettre (dont la signature alignée à droite de la page est le premier mais non le seul élément) est encore plus délicate, car elle ne se manifeste jamais par un élément visuellement massif comme un titre.

En raison de cette complexité, nous avons opté pour une typologie de régions resserrée. Tandis que l'ontologie de F. Chiffoleau proposait un type de région pour chaque

16. C'est aussi un enjeu central du projet **Galli(corpor)a**

17. 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).

18. 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).

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

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

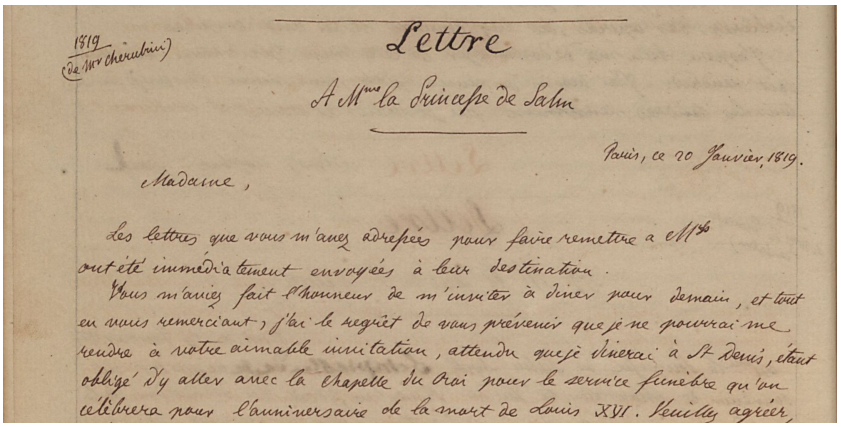


FIGURE 1.1 – Début d'une lettre présentant une disposition aérée des éléments (SALM (Constance de), *Correspondance générale, seconde copie, 2e volume, 1815-1821*, URL : `https://constance-de-salm.de/archiv/#/document/11216` (visité le 11/04/2022)).

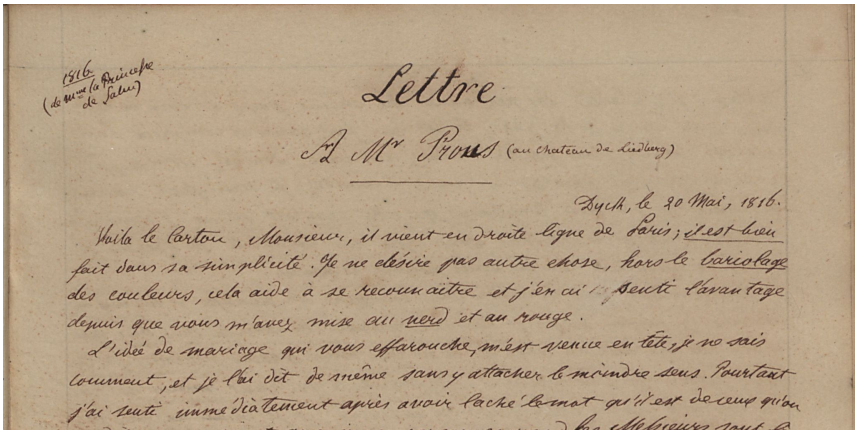


FIGURE 1.2 – Début d'une lettre présentant une disposition resserrée des éléments (SALM (Constance de), *Correspondance générale, seconde copie, 2e volume, 1815-1821*, URL : `https://constance-de-salm.de/archiv/#/document/11216` (visité le 11/04/2022)).

élément significatif de l'écrit de correspondance [21], nous n'avons retenu que trois types de régions principaux pour annoter la structure des lettres :

— **MainZone** : pour le corps du texte ;
— **CustomZone :opener** : pour l'en-tête des lettres ;
— **CustomZone :closer** : pour la clôture des lettres.

la définition des régions est formelle, visuelle

1.2.2 Lignes d'écriture

Les types de lignes dont on propose l'utilisation sont :

21. F. Chiffoleau, *[Correspondance En Langue Française, XXe s.]…*

1.2.3 Phénomènes graphiques particuliers

C. de Salm a corrigé certains mots de sa main :

— En rayant une lettre, un mot ou plusieurs mots, ou bien en réécrivant par dessus le texte. Dans de nombreux cas cela consiste en une simple lettre barrée ; le typage de la ligne demanderait alors beaucoup d'effort pour un résultat minime ;

— En réécrivant dans l'interligne : il est alors pertinent d'utiliser le type de ligne e-Scriptorium **Correction**.

Un ensemble de solutions d'encodage des corrections a été proposé dans le cadre du projet DAHN [22]. J'envisage plutôt **ne pas encoder ces éléments dans la phase d'HTR**, et de ne les aborder que la phase d'édition. Il sera de toute façon necessaire, lors de la reprise manuelle de l'édition TEI, de suivre la reproduction du manuscrit à éditer. En outre, introduire des caractères tels que £, €, etc. dans la transcription génèrerait du bruit dans l'entraînement du modèle HTR et imposerait une phase de nettoyage pour les réutilisations éventuelles des vérités de terrain.

En somme, il s'agirait de **transcrire tout ce qui est lisible** (y compris les lettres biffées, lorsque c'est possible), en privilégiant le dernier état du texte dans le cas où la correction a été superposée à la première couche d'écriture.

1.2.4 Entraîner des modèles de segmentation des pages

Cette section est à écrire.

1.2.5 Contrôler la pertinence de la segmentation

Le script python écrit pour permettre l'importation sélective des images inventoriées [23] permet en outre de contrôler l'association entre les images sélectionnées et les notices de l'inventaire. Plusieurs lettres peuvent en effet être inventoriées pour la même image, mais surtout une image peut contenir un mélange de lettres inventoriées et de lettres non inventoriées.

Or, il est crucial de pouvoir contrôler le statut de chaque lettre présente dans l'image. Les lettres non-inventoriées étant par définition absentes des données de l'inventaire, il n'existe aucun moyen, en aval de la transcription automatisée des textes pour sélectionner les transcriptions pertinentes (celles des lettres inventoriées) des transcriptions non pertinentes (lettres non inventoriées). Ainsi, il n'est possible de gérer correctement la transformation des transcriptions en édition qu'en ayant préalablement exclu toutes les parties de texte non pertinentes, un travail qui ne peut être automatisé.

22. Id., *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).

23. S. Biay, *donneesImages.Py…*

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

1.3 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 démarche [24].

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 copies [25]. Il était dès lors impossible d'envisager entraîner des modèles particuliers pour chaque écriture en découpant 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 suivre [26].

---

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

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

26. 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` (visité le 10/06/2022). Une partie de cette méthodologie a été présentée dans le cadre de la réunion mensuelle du DHIP : S. Biay et Pauline Spychala, « L'intelligence Artificelle à l'IHA », dans *Hausinfo*, Paris, IHA, 2022.

1.3.1 Sélectionner des échantillons d'écriture et organiser les fichiers

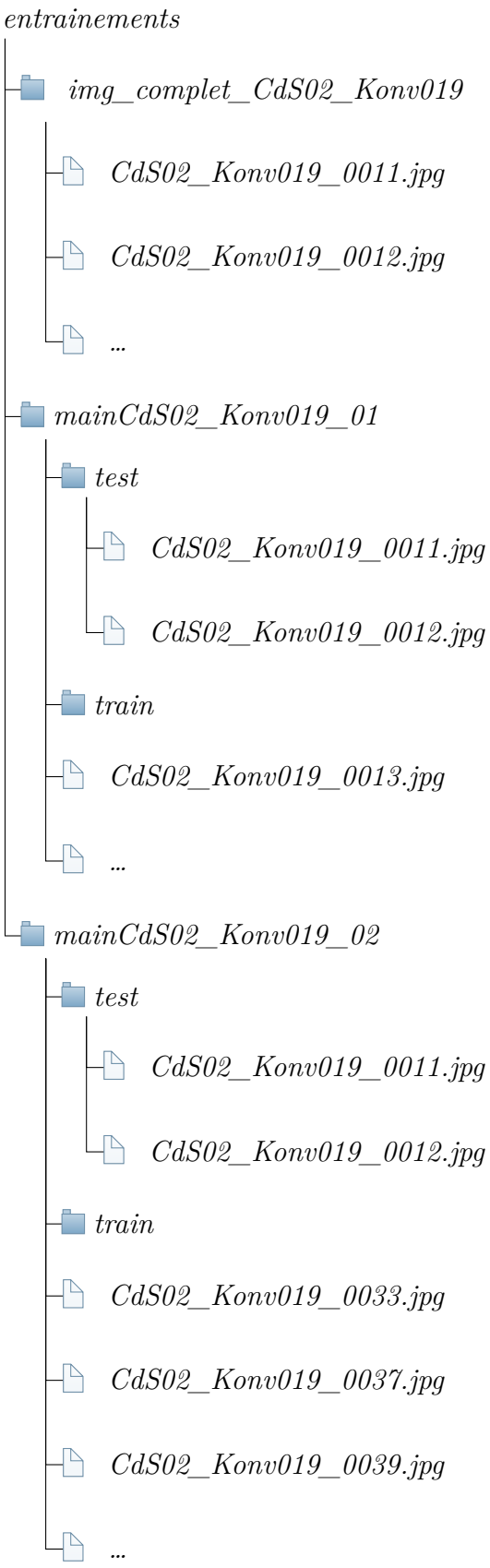
Entraîner des modèles à reconnaître les écritures de plusieurs mains différentes suppose un regard attentif aux variations paléographiques, mais aussi une grande rigueur de gestion des fichiers et de leurs données, car il s'agit d'abord de classer par type d'écriture les reproductions photographiques 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 loin [27].

27. Cf. *infra* 1.3.6, p. 22.


entrainements
img_complet_CdS02_Konv019
CdS02_Konv019_0011.jpg
CdS02_Konv019_0012.jpg
...
mainCdS02_Konv019_01
test
CdS02_Konv019_0011.jpg
CdS02_Konv019_0012.jpg
train
CdS02_Konv019_0013.jpg
...
mainCdS02_Konv019_02
test
CdS02_Konv019_0011.jpg
CdS02_Konv019_0012.jpg
train
CdS02_Konv019_0033.jpg
CdS02_Konv019_0037.jpg
CdS02_Konv019_0039.jpg
...

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 transcriptions [28].

1.3.2 Établir des normes de transcription

Il faut évoquer brièvement ici les principes généraux de la transcription des textes, les normes détaillées étant reportées en annexe [29].

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 lettres [30]), soit comme une transcription déjà interprétative de la source qui uniformise les allographes

28. 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*

29. Cf. A, p. 31.

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

et restitue les abréviations [31]. En réalité, aucune de ces options ne peut être appliquée de façon radicale de bout en bout d'une transcription, chacune rencontrant des limites dans son applicabilité.

Par exemple, le projet Notre-Dame de Paris et son cloître (e-NDP) aborde l'entraînement des algorithmes de reconnaissance d'écriture dans l'optique de leur apprendre à restituer les abréviations des scribes des sources du chapitre [32]. 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 abrégé [33].

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'une espace (à partir de quelle quantité de blanc une espace doit être transcrite) 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* [34]. 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 génériques prônés par la communauté scientifique consistuée autour du projet Cremma et qui correspondent de manière tendantielle au concept de transcription *graphématique* traduit et expliqué par D. Stutzmann [35].

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

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

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

33. 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.

34. A. Pinche, *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^o^ 3*, CREMMALAB, 2021, URL : `https://cremmalab.hypotheses.org/compte-rendu-seance-3` (visité le 13/06/2022).

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

culté et le coût impliqué par l'imitation de l'espacement des mots a été tranchée par la restitution de l'espacement moderne des mots dès lors que l'usage scribal ne caractérisait pas un usage établi ; en revanche, les élisions, agglutinations ou encore les lexicalisations (consacrées ou fautives) ont été respectées : *d'avantage, Ç'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, on a appliqué les conventions préconisées par la convention de Leyde [36], retenues dans le cadre du Cremma [37].

1.3.3 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 travail [38]. 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 XML-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.

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

37. A. Pinche, *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).

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

1.3.4 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 commande [39] (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. Souvay [40], les performances de ce dernier ont été évaluées à partir de transcriptions nouvellemement produites. L'acuité de reconnaissance de l'écriture sur l'unique main attestée dans le corpus d'entraînement de ce dernier a atteint 77,25% seulement.

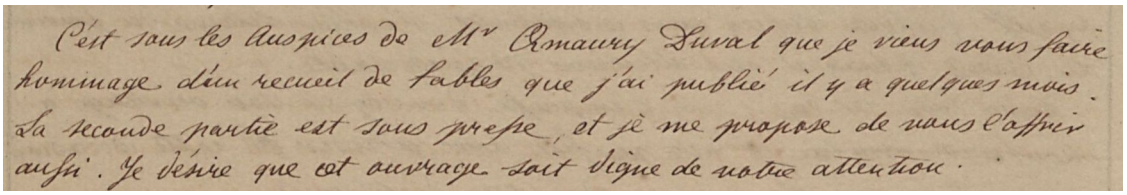


FIGURE 1.3 – Copie d'une lettre de Pierre-Augustin Rigaud à C. de Salm, le 13 avril 1824.

Voici la prédiction correspondante :

> cet ju le gusaires de Mr touaauz ctral ne ; qus venes tat
> bonmage Liu rececit de fables que j'li puslié il y a quelsurs mos
> L ronde partie est sous presse. et je me prapose de vous l'affrir
> auissi. Je dhrre qe cet ouvrage sait digne de votre atenson [41]

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

> Cest saus les Auopices de Mr̂ Amaury Duval rue je vieus mous favre hommage
> dem recueit de lables que ai publie et & a quetques mais la secoude 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

39. *Kraken [Documentation]*...

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

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

42. 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.

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'oeil 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.

En procédant à la constitution d'une vérité de terrain d'une dizaine de pages 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 lettres [43] : 98,68%
— 3e main de la seconde copie des lettres [44] : 96,31%
— 1re main de la correspondance Martini [45] : 97,88%
— 2e main de la correspondance Martini [46] : 96,27%

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

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

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

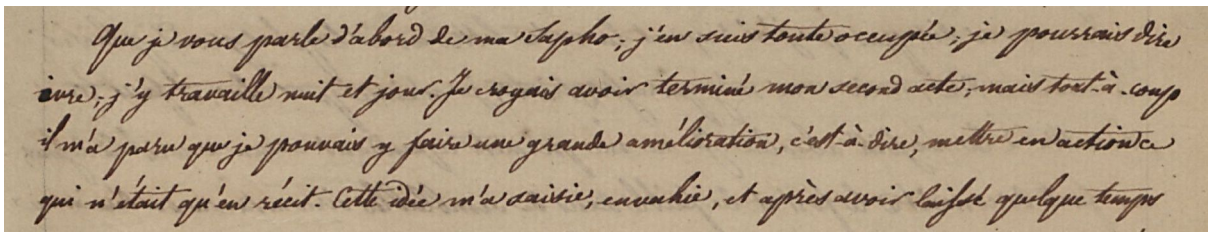


FIGURE 1.4 – Copie d'une lettre adressée par C. de Salm à Jean-François Thurot, le 21 février 1794.

> Que j vous parle d'abord le ma Saphe ; j'en suis toute occuple je pourrais drie
> êvre j'y travaille nuit et jour. Jecrogais avoir terminé mon second acte, mais toet à-coup
> Ima 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 tempps [47]

43. Dénommée mainCdS02_Konv002_01.
44. Dénommée mainCdS02_Konv002_03.
45. Dénommée mainCdS02_Konv019_01.
46. Dénommée mainCdS02_Konv019_02.
47. Notice d'inventaire : *CdS/02_1/031-032*, Die Korrespondenz der Constance de Salm (1767-1845).

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).

1.3.5 Transcription manuelle *versus* trasncription automatique : quelle bonne méthode pour l'entraînement ?

Cette section est à écrire.

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

Les performances du nouveau modèle, dénommé cds_lectcm_04_mains_01 [48], 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 résultats [49].

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 modèle [50].

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, URL : `https://constance-de-salm.de/archiv/#/document/8440` (visité le 13/06/2022).

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

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

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

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 entraînement que l'on a effectué à partir d'un corpus plus riche. Cette option est celle qui garantit la plus grande généricité 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 généricité 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.

1.3.7 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 XML-Alto contenant les transcriptions manuelles aux fichiers contenant la prédiction automatique du texte des autres pages d'un même dossier est de regrouper ces fichiers ensemble. Or, nous avons voulu tenir compte de la possibilité que les transcriptions manuelles ne recouvrent pas toutes les lignes d'écriture d'une page le cas n'est pas très fréquent, mais nous y avons été confronté. Certaines mains n'étant attestées que de manière sporadique, en compagnie d'autres écritures, la méthodologie d'entraînement impose de ne transcrire que l'écriture propre au test ou à l'entraînement, laissant les écritures voisines de côté. Il résulte de cette nécessité que les fichiers XML-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 XML-Alto est rendu possible par l'identifiant unique de chaque élément contenant une ligne de texte (TextLine). Nous avons donc écrit un script python [51] capable d'analyser toutes les lignes d'écriture des fichiers de nos vérités de terrain et de comparer

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

leur identifiant avec ceux des lignes des fichiers des prédictions automatiques portant les mêmes noms. En cas de correspondance entre les identifiants, la transcription manuelle vient remplacer la prédiction du texte.

1.4 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.

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

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

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

Nous avons suivi la démarche explicitée dans la documentation du projet DAHN [52] et proposé quelques développements aux scripts issus de ce projet [53].

52. F. Chiffoleau, *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).

53. Le script principal porte le nom de *spellcheckTexts* : S. Biay et F. Chiffoleau, *spellcheckTexts.Py*, 6 avr. 2022, URL : `https://github.com/sbiay/CdS-edition/blob/main/htr/py/spellcheckTexts.py`

1.4.1 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. L'automatisation des corrections ne permet absolument pas de faire l'économie d'une révision approfondie du texte par l'éditeur. Elle doit par conséquent faire preuve d'un haut degré de performance : son but est d'abord et avant tout de faire gagner du temps. Or, chaque forme signalée au cours de cette phase requiert une décision de l'éditeur : ces formes doivent donc être très pertinentes afin de ne pas gaspiller le temps de ce dernier. Contrôler chaque mot dans son contexte serait largement contre productif.

Il est donc très vite apparu nécessaire de ne pas signaler à l'éditeur les mots dont la graphie a été validée par ailleurs. Pour cela, on s'est appuyé d'une part sur un dictionnaire généraliste de la langue française et d'autre part sur les mots de la correspondance-même de C. de Salm, à savoir les mots contenus dans les vérités de terrain que l'on a produites pour le test et l'entraînement des modèles de reconnaissance d'écriture [54].

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

1.4.2 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*, mais n'est pas capable d'associer la forme *pubiès* aux mots de la famille de *publier*.

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

(visité le 19/04/2022). Ce script est fondé sur l'utilisation du module publié par Tyler Barrus, *Pyspellchecker : Pure Python Spell Checker Based on Work by Peter Norvig*, version 0.6.3, URL : `https://github.com/barrust/pyspellchecker` (visité le 19/04/2022).

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

Une fois les corrections validées, un second script écrit par F. Chiffoleau permet de les appliquer aux fichiers contenant les textes [55]. Originellement conçu pour remplacer des chaînes de caractère n'importe où dans le fichier concerné, il faisait courir le risque de remplacements abusifs. Par exemple, si la forme *natur* devait être corrigée en *nature* et que la même page de texte contenait aussi le mot *naturellement*, une application globale des corrections entraînerait la création d'une faute : *naturellement* deviendrait *natureellement*. Le script a donc été perfectionné afin de procéder à l'application des corrections ligne par ligne et mot par mot [56]

En outre, il s'est avéré nécessaire de modifier la méthode d'application des corrections aux fichiers XML-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'origine [57].

1.4.3 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*).

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

Cette méthode présentait plusieurs inconvénients :

— Une fois que l'on avait procédé à des corrections pour les mots d'une page, le script qui les intégrait au fichier rassemblant toutes les corrections contrôlait qu'une forme ne puisse pas être associée à plusieurs corrections. Lorsqu'une ambiguïté était repérée, il fallait intervenir sur les deux fichiers pour neutraliser la correction. Devenu fréquent, ce processus diminuait le bénéfice de temps attendu de la correction automatique ;

— D'autre part, il s'est avéré que les corrections ambiguës sont nombreuses, car il

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

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

57. L'injection des transcriptions manuelles en lieu et place des prédictions (cf. ci-dessus, 1.3.7, p. 23) dans les seuls fichiers appartenant au corpus d'entraînement de la reconnaissance d'écriture a entraîné une modification irrémédiable de l'indentation de ceux-ci. L'indentation de ces fichiers étant devenue différente des autres fichiers des prédictions, il n'était plus possible de s'appuyer sur l'identité des indentations pour repérer les lignes de textes à remplacer. Il devenait donc obligatoire de s'appuyer sur la hiérarchie de l'arbre XML pour appliquer ces corrections.

suffit d'une faute sur un petit mot pour le rendre ambigu avec un autre mot : *uue* peut être corrigé en *rue* ou en *une* ; *veus* peut être corrigé en *veux* ou en *vous* ; *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 écrite [58] 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.

1.4.4 Élaborer et enrichir un nouveau dictionnaire de la langue française

58. Il s'agit de la fonction dénommée *ordreOccurrences* ; cf. Id., *spellcheckTexts.Py…*

Annexes

Annexe A

Normes de transcription

A.1 Accentuation

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

A.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.

A.3 Séparation des mots

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

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

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

A.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. Prons* pour *Mr. Prous*.

A.5 Abréviations

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

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

A.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 .

A.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 Leyde [1], retenues dans le cadre du Cremma [2].

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)*.

A.8 Passages illisibles

Pour les problèmes de déchiffrement du texte, la convention de Leyde n'a pas d'autre préconisation que la mention en apparat [3].

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 »…).

Bibliographie

Scripts

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).

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

— *injectTranscript.Py*, 20 mai 2022, URL : `https://github.com/sbiay/CdS-edition/blob/main/htr/py/injectTranscript.py` (visité le 03/06/2022).

— *journalReconn.Py*, 20 mai 2022, URL : `https://github.com/sbiay/CdS-edition/blob/main/htr/py/journalReconn.py` (visité le 25/05/2022).

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

Biay (Sébastien) et Chiffoleau (Floriane), *spellcheckTexts.Py*, 6 avr. 2022, URL : `https://github.com/sbiay/CdS-edition/blob/main/htr/py/spellcheckTexts.py` (visité le 19/04/2022).

— *textCorrection.Py*, 6 avr. 2022, URL : `https://github.com/sbiay/CdS-edition/blob/main/htr/py/textCorrection.py` (visité le 19/04/2022).

Valorisation du projet

Biay (Sébastien) et Spychala (Pauline), « L'intelligence Artificelle à l'IHA », dans *Hausinfo*, Paris, IHA, 2022.