

Lucas Terriel

Licencié en histoire

Diplômé de master Histoire, Civilisations, Patrimoine contemporains

Représenter et évaluer les données issues du traitement automatique d'un corpus de documents historiques

L'exemple de la reconnaissance des écritures manuscrites dans
les répertoires de notaires du projet LectAuRep

Mémoire pour le diplôme

« Technologies numériques appliquées à l'histoire »

2020





LICENCE OUVERTE
OPEN LICENCE

Document placé sous les termes de la licence ouverte 2.0.

Le « Concédant » concède au « Réutilisateur » un droit non exclusif et gratuit de libre « Réutilisation » de l’« Information » objet de la présente licence, à des fins commerciales ou non, dans le monde entier et pour une durée illimitée, dans les conditions exprimées ci-dessous.

Le « Réutilisateur » est libre de réutiliser l’« Information » :

- de la communiquer, la reproduire, la copier ;
- de l’adapter, la modifier, l’extraire et la transformer, notamment pour créer des « Informations dérivées » ;
- de la diffuser, la redistribuer, la publier et la transmettre, de l’exploiter à titre commercial, par exemple en la combinant avec d’autres informations, ou en l’incluant dans votre propre produit ou application..

Sous réserve de :

- mentionner la paternité de l’« Information » : sa source (a minima le nom du « Concédant ») et la date de la dernière mise à jour de l’« Information » réutilisée.
- Le « Réutilisateur » peut notamment s’acquitter de cette condition en indiquant l’adresse (URL) renvoyant vers « l’Information » et assurant une mention effective de sa paternité.
- Cette mention de paternité ne doit ni conférer un caractère officiel à la « Réutilisation » de l’« Information », ni suggérer une quelconque reconnaissance ou caution par le « Concédant », ou par toute autre entité publique, du « Réutilisateur » ou de sa « Réutilisation ».
- Le « Réutilisateur » est seul responsable de la « Réutilisation » de l’« Information ».

Consulter la licence en entier pour plus de détails :

[version française](#) | [english version](#)

Résumé

Ce mémoire a été réalisé dans le cadre de l'obtention du diplôme de Master 2^{ème} année « Technologies numériques appliquées à l'histoire » de l'École nationale des chartes. Il est rédigé dans le contexte d'un stage de quatre mois au sein de l'équipe projet ALMAnaCH de INRIA, et dont l'action s'est inscrite dans le cadre de la phase 3 du projet Lectaurep, porté par le Ministère de la Culture, les Archives Nationales, et Scripta portant sur la reconnaissance des écritures manuscrites et l'extraction d'informations sur les répertoires de notaires (1803-1944). Le projet met en œuvre les méthodes récentes d'apprentissage machine qui utilisent la plate-forme de transcription *eScriptorium*, interface graphique du système HTR à base de réseaux de neurones artificielles *Kraken*. Il parcourt les tentatives de constitution d'un fichier pivot XML-TEI afin de structurer les données entrantes et sortantes de la plate-forme *eScriptorium*, notamment lors de la récupération d'images. Il examine ensuite les étapes de recherche de métriques pour comparer des chaînes de texte et du développement d'une application Python dédiée à l'évaluation des modèles de transcription *Kraken-Benchmark*, testée sur les données de Lectaurep. Ces métriques pouvant être intégrées dans le *workflow* général du projet et dans le fichier pivot XML-TEI. Ce mémoire s'attache à montrer comment ces axes pourraient être améliorés par la suite. Il s'agit d'une présentation des stratégies, des choix critiques et des enjeux envisagés durant le stage, dont l'objectif est de rendre compte des réalisations techniques et des choix scientifiques opérés dans un contexte mêlant le patrimoine et les enjeux numériques.

Mots-clefs : Lectaurep ; Archives nationales ; Minutier central ; INRIA ; Répertoires de notaires ; XIX^e siècle ; XX^e siècle ; HTR ; Format pivot ; XML ; TEI ; ALTO ; EAD ; EAC ; EXIF ; ODD ; XSLT ; Développement applicatif ; Python ; métriques ; Intelligence artificielle ; Réseaux de neurones ; *data science* ; Apprentissage machine ; Similarité syntaxique ; Similarité sémantique ; TAL ; Humanités numériques

Informations bibliographiques : Lucas Terriel, *Représenter et évaluer les données issues du traitement automatique d'un corpus de documents historiques. L'exemple de la reconnaissance des écritures manuscrites dans les répertoires de notaires du projet LectAuRep.*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Alix Chagué et Thibault Clérice, École nationale des chartes, 2020.

Remerciements

Pour ce stage, qui s'est déroulé durant la période de confinement, je tiens à remercier l'ensemble des personnes qui m'ont aidé et soutenu dans cette situation particulière et qui m'ont encouragé dans mon travail.

Je remercie ma tutrice professionnelle, Mme Alix Chagué, et mon tuteur pédagogique, M. Thibault Clérice, pour leurs appuis et leurs conseils avisés (Merci Alix pour tes conseils « pythoniques »...).

Je remercie l'équipe ALMAAnaCH d'Inria, qui a su créer des conditions favorables d'accueil pour mon stage et a permis un environnement de travail stimulant ; je remercie tout particulièrement M. Laurent Romary, directeur de recherche, pour nos échanges et conseils pertinents sur la TEI et Mme Florianne Chiffoleau, ingénieure recherche et développement, pour son aide.

Comme le stage est un travail d'équipe, je remercie Jean-Damien Généro, collègue du master TNAH à l'École nationale des chartes, qui était également stagiaire sur la même période à ALMAAnaCH sur le projet « Time us ». Nos échanges et la mise en commun de nos travaux ont contribué à rendre le stage encore plus enrichissant.

Je remercie l'ensemble de l'équipe du projet Lectaurep, et le personnel du Minutier central des notaires de Paris aux Archives nationales pour leur disponibilité : Mme Marie-Françoise Limon-Bonnet, responsable du département, Mme Aurélia Rostaing, responsable du pôle instruments de recherche, M. Gaetano Piraino, responsable à la DMOASI, M. Danis Habib, chargé d'études documentaires, Mme Virginie Grégoire, secrétaire de documentation, M. Benjamin Davy, agent technique d'accueil, surveillance et magasinage, et Mme Anna Chéru, stagiaire en phase 3.

Enfin je remercie, ma famille, mes camarades de promotions : Duchesse Anne Brunet, Mathilde Daugas, Edward Gray PhD, Chloë Fize (11101010 10011110 10101001), Morgane Rousselot ; mes amis de longue date : Léa Mièle et Benjamin Luzu, pour leur indéfectible soutien tout au long de cette période.

Liste des sigles et abréviations

* ORGANISMES *

- AN : *Archives Nationales*
- ALMANACH : *Automatic Language Modelling and Analysis & Computational Humanities*
- DMC : *Département du Minutier central des notaires de Paris*
- DMOASI : *Département de la maîtrise douvrage du système d'information (direction de l'appui scientifique)*
- INRIA : *Institut Nationale de Recherche en Informatique et Automatique*
- W3C : *World Wide Web Consortium*

* DOMAINES ET DISCIPLINES *

- IA : *Intelligence Artificielle*
- DL : *Deep Learning*
- ML : *Machine Learning*
- POO : *Programmation Orientée Objet*
- REN : *Reconnaissance d'Entités Nommées* (en anglais, NER ou *Name Entity Recognition*)
- RNN : *Recurrent neural network* (en français, *Réseau de neurones récurrents*)
- TAL : *Traitemet Automatique des Langues* (en anglais, NLP ou *Natural Language Processing*)

* TECHNOLOGIES *

- API : *Application Programming Interface* (en français, *interface de programmation d'application*)
- BASH : *Bourne-Again shell*
- CMS : *Content Management System* (en français, *système de gestion de contenu*)
- CSS : *Cascading Style Sheets*

- CSV : *Comma-separated values*
- DTD : *Document Type Definition*
- HTML : *HyperText Markup Language*
- HTTP : *Hypertext Transfer Protocol*
- HTR : *Handwritten Text Recognition*
- IIIF : *International Image Interoperability Framework*
- JSON : *JavaScript Object Notation*
- JSON-LD : *JavaScript Object Notation for Linked Data*
- OCR : *Optical Character Recognition*
- ODD : *One Document Does it all*
- OS : *Operating System* (en français, *Système d'exploitation*)
- PDF : *Portable Document Format*
- RDF : *Ressource Description Framework*
- RELAXNG : *Regular Language for XML Next Generation*
- SGML : *Standard Generalized Markup Language*
- TEI : *Text Encoding Initiative*
- WSGI : *Web Server Gateway Interface*
- XML : *eXtensible Markup Language*
- XSLT : *eXtensible Stylesheet Language Transformations*

*** STANDARDS, NORMES, RÉFÉRENTIELS ET ONTOLOGIES ***

- ALTO : *Analysed Layout and Text Object*
- EAC-CPF : *Encoded Archival Context - Corporate Bodies, Persons and Families*
- EAD : *Encoded Archival Description*
- EXIF : *EXchangeable Image File Format*
- ISAD(G) : *International Standard Archival Description-General* (en français, *Norme générale et internationale de description archivistique*)
- ISAAR(CPF) : *International Standard Archival Authority Record for Corporate Bodies, Persons and Families* (en français, *Norme internationale sur les notices d'autorité archivistiques relatives aux collectivités, aux personnes et aux familles*)
- RIC-O : *Records in Contexts-Ontology*
- RIC-CM : *Records in Contexts-Conceptual Model*
- EXIF : *EXchangeable Image File Format*

— TEI : *Text Encoding Initiative*

★ MÉTRIQUES ★

- CER : *Character Error Rate* (en français, *Taux d'erreur de caractères*)
- WACC : *Word Accuracy* (en français, *Taux de reconnaissance de mots*)
- WER : *Word Error Rate* (en français, *Taux d'erreur de mots*)

★ AUTRES ★

- EN : *Entités Nommées*
- PEP : *Python Enhancement Proposals*
- SIV : *Salle des inventaires virtuels*
- URI : *Uniform Resource Identifier* (en français, *identifiant uniforme de ressource*)
- URL : *Uniform Resource Locator* (en français, *localisateur uniforme de ressource ou adresse web*)

Introduction

Introduction

Depuis la caractérisation de certains types d'écritures par le moine bénédictin mauriste Jean Mabillon (1632-1707) publiée dans son traité de diplomatique *De re diplomatica* en 1681, considéré comme un temps fondateur pour la paléographie, en passant par la traduction des hiéroglyphes en 1822 par Jean-François Champollion (1790-1832) ; lire, comprendre et déchiffrer les écritures pour analyser son passé et envisager l'avenir est une constante chez l'homme. Au début du XXI^e siècle, un nouveau cap a été franchi dans la tentative de décoder et de présenter les sources anciennes de l'histoire avec l'émergence des technologies numériques.

D'un côté les institutions patrimoniales ont massifiée la numérisation et la mise en ligne de leurs collections sous la forme de bases de données qui continuent de croître ; de l'autre la discipline informatique a envisagé le moyen de doter les machines d'une « intelligence » proche des capacités humaines, comme la possibilité de voir, de lire et de reproduire un texte, cela rendu possible par l'apprentissage des réseaux de neurones artificiels. Enfin les sciences humaines et sociales, tentent de comprendre et de tirer parti de ces innovations afin d'envisager d'autres approches méthodologiques sur les sources. Depuis 2018, le projet Lectaurep (LECTure AUtomatique de REPertoires) se situe à la confluence de ces trois domaines en essayant de faire dialoguer les archivistes, les ingénieur(e)s en informatique et les chercheuses et les chercheurs en sciences humaines pour trouver de nouveaux moyens de rendre les fonds du département du Minutier central des notaires parisiens des Archives nationales exploitables à distance pour des publics désormais « familiers » des outils numériques.

Le projet Lectaurep¹, qui est entré dans sa troisième phase en novembre 2019, vise à repenser l'usage qui est fait actuellement des répertoires de notaires, l'une des sources du Minutier central les plus consultées. Alliant des domaines de l'intelligence artificielle comme la reconnaissance d'écriture manuscrite (HTR) et la recherche d'informations (NER), Lectaurep utilise la plate-forme *eScriptorium*² (PSL/eScripta), qui est basée sur le logiciel OCR/HTR appelé *Kraken*³. Les applications à terme de la vision par ordinateur sont nombreuses pour les archives : recherche en texte intégral et extraction d'informations dans plus de deux milles répertoires comptant trois cents à cinq cents pages chacun, outil de transcription collaboratif, outils d'analyses quantitatives (fiscalité notariale) et de visualisation de l'activité géographique des notaires, édition numérique des répertoires etc. Les Archives nationales (Ministère de la Culture) se sont associées à Inria et à son équipe-projet ALMAAnaCH dans le cadre d'une convention-cadre pour mettre en oeuvre ces technologies.

1. Projet Lectaurep, Archives nationales, URL : <http://www.archives-nationales.culture.gouv.fr/1-intelligence-artificielle-et-le-patrimoine>

2. *eScriptorium*, URL : <https://escripta.hypotheses.org/>

3. *Kraken*, URL : <http://kraken.re/>

Inria, anciennement Institut national de recherche en informatique et automatique, est un établissement public à caractère scientifique et technologique placé sous la double tutelle du ministère de l’Enseignement supérieur, de la Recherche et de l’innovation et du ministère de l’Économie et des Finances. Créé le 3 janvier 1967 à l’occasion du « plan calcul »⁴, elle accompagne la recherche en informatique et en mathématiques et crée des partenariats privés ou publics assurant les transferts de technologies. Depuis 2018, Inria fait figure de plate-forme stratégique dans le développement de l’intelligence artificielle. À côté de ces missions, elle assure la valorisation des sciences de l’information et de la communication (Mooc), monte des *startups* technologiques et édite des logiciels *open-source*⁵. Inria compte trois mille cinq cents chercheurs, chercheuses et ingénieur(e)s répartis en deux cents équipes-projet dans huit centres de recherche axés sur des domaines scientifiques tels que la science des données, le calcul haute performance, l’intelligence artificielle, l’informatique théorique, la sécurité informatique etc. ALMAaCH (*Automatic Language Modelling and Analysis & Computational Humanities*) est l’une des équipes-projet de Inria spécialisée dans le développement de logiciels et de ressources en traitement automatique du langage naturel (TAL) en s’appuyant sur les méthodes récentes d’apprentissage profond. L’équipe accompagne les chercheurs, les chercheuses et les institutions patrimoniales dans les transitions technologiques qu’impliquent le récent paradigme des humanités computationnelles⁶. L’équipe est coordonnée par le responsable scientifique, Benoît Sagot, et des membres permanents, Laurent Romary, Djamel Seddah, Éric Villemonte de La Clergerie et Rachel Bawden.

Le présent mémoire rend compte du stage qui s’est déroulé du 30 mars 2020 au 31 juillet 2020 au sein de l’équipe ALMAaCH d’Inria pour le projet Lectaurep. Cependant, le contexte lié à l’épidémie de COVID-19, m’a obligé, comme la plupart des stagiaires et des salarié(e)s à effectuer la totalité de mon stage à distance avec comme principale interlocutrice Alix Chagué, ingénierie en recherche et tutrice de stage. Mon intérêt pour les archives m’avait poussé à effectuer un stage en juin 2019 au Minutier central des Archives nationales où j’avais découvert les fonds ainsi que le projet Lectaurep. L’envie de mettre à profit mes compétences informatiques acquises tout au long de l’année ainsi que de découvrir des aspects de développement plus poussés dans les domaines de la *Datascience* et du TAL m’a conforté dans le choix de ce stage.

4. « Notre histoire | Inria », URL : <https://www.inria.fr/fr/notre-histoire>

5. « Dix logiciels stars d’Inria | Inria », URL : <https://www.inria.fr/fr/dix-logiciels-stars-dinria>

6. ALMAaCH-Équipe-projet Inria, URL : <https://team.inria.fr/almanach/fr/>

Introduction

Le projet Lectaurep, qui a connu trois phases d'essais, s'appuie sur des outils en cours de développement comme la plate-forme *eScriptorium*, qui demandent un soutien régulier pour maintenir les principales fonctionnalités. Dans ce contexte d'expérimentation des outils et dans la recherche d'innovations ; en dehors d'un cahier des charges à mettre en oeuvre et d'une mise en production imminente d'une chaîne de traitement arrêtée, j'ai cherché à savoir comment mettre en place des outils et des formules pour représenter, structurer et évaluer les données manipulées jusqu'à présent dans Lectaurep.

À côté des missions portant sur la valorisation du projet et la prise en main de *Kraken* par l'intermédiaire d'entraînements de modèles de transcriptions, mes missions principales étaient les suivantes :

- Réfléchir à la mise en place d'un format pivot XML-TEI pour structurer les données lors de l'import des images et de l'export des transcriptions de vérités terrains dans *eScriptorium*, tout en soulignant les avantages de la TEI dans la poursuite du projet.
- Développer un outil généralisable à d'autres projets HTR, nécessaire dans la suite du projet Lectaurep, pour évaluer les modèles de transcription et tester cet outil sur des jeux de données Lectaurep.

Si nous reviendrons, dans un premier temps, sur les aspects historiques et technologiques inhérents au projet Lectaurep, c'est pour mieux appréhender les types de données, métadonnées et les contraintes posées par les documents et les Archives nationales. Cependant, dans un second temps, nous verrons que la mise en place d'une première version d'un fichier XML-TEI pivot peut permettre de structurer ces métadonnées et de les récupérer lors des imports et des exports au sein de la plate-forme *eScriptorium*. Dès lors, dans une dernière partie, nous aborderons la création *ex-nihilo* d'un outil en Python, *Kraken-Benchmark*, pour permettre l'évaluation et l'analyse des modèles HTR à partir de métriques. Les résultats pouvant être intégrés au fichier pivot XML-TEI et/ou dans la chaîne de traitement globale du projet.

Première partie

**Le projet Lectaurep : un cas
d'application de l'« intelligence
artificielle » aux documents
historiques**

Chapitre 1

Lectaurep, un projet de recherche et développement en analyse et reconnaissance de document

1.1 Lectaurep, un enfant né de la rencontre du numérique et du patrimoine

1.1.1 La transformation digitale

Le projet Lectaurep s'inscrit dans le temps long de la rencontre du numérique et des institutions patrimoniales. L'application des potentialités qu'offre l'intelligence artificielle aux projets patrimoniaux, est l'aboutissement d'efforts menés conjointement par les institutions culturelles et universitaires, pour rejoindre la « transformation digitale ». Cette dernière à touché l'ensemble des secteurs de l'économie et de l'industrie. Elle prend véritablement son essor avec la démocratisation de la micro-informatique et l'apparition du *World Wide Web (web)*, inventé entre 1989 et 1990 par Tim Berners-Lee, Jean-François Abramatic et Robert Cailliau.

Le chercheur en humanités numériques, Milad Douehi, qualifie même de « grande conversion numérique » ce bouleversement des pratiques de recherche. Cette étape est considérée comme un nouveau processus civilisateur (terme emprunté au sociologue Norbert Elias) au même titre que l'évolution des bonnes manières et des mœurs en Europe depuis le Moyen-Âge :

L'environnement numérique, tout en offrant un meilleur accès à l'information et, dans certains cas, des libertés bien nécessaires, introduit des modes nouveaux et puissants de surveillance et de censure. Autrement dit, par son mode de fonctionnement, la culture numérique ressemble beaucoup à un processus civilisateur, qui apporte avec lui de nouvelles possibilités mais aussi des effets secondaires imprévisibles et parfois inquiétants, voire dangereux.¹

Les institutions culturelles se sont confrontées aux nouvelles pratiques des usagers et aux problématiques qu'elles posent, notamment, rendre accessible leurs collections par l'intermédiaire des nouvelles technologies.

1.1.2 Le mouvement de la numérisation des institutions patrimoniales dans les années 1990

Dès les années 1990, les services d'archives, musées et bibliothèques ont pressenti les avantages de mener à bien des **politiques de numérisation massives des documents**, afin d'offrir un accès plus large à leurs publics. La Bibliothèque du Congrès des États-Unis fut l'une des premières à initier le mouvement avec son projet « American Memory ». La bibliothèque numérise des livres, des affiches, des enregistrements sonores et des photographies, retracant l'histoire et la culture des États-Unis. Cette plate-forme compte actuellement plus de neuf millions de documents numérisés².

En France ce mouvement est suivi de près par la Bibliothèque Nationale de France (BNF) qui met en place, en 1998, le projet « Gallica »³. Elle propose ainsi de mettre à disposition des documents libres de droits suivant des thématiques (Grande Guerre, sport, etc), et représente aujourd'hui plus de deux millions de documents⁴.

Dans le domaine des archives, ce sont les services d'archives départementaux qui sont les premiers à commencer à rendre leurs contenus accessibles en ligne directement sur leurs sites institutionnels. Ils passent alors du statut de « sites "vitrines" statiques, à de véritables salles de lecture virtuelles, permettant la consultation d'instruments de recherche et de documents d'archives numérisés »⁵. L'ambition de ces institutions était

1. Milad Doueihi, *La Grande conversion numérique. suivi de Rêveries d'un promeneur numérique*, Seuil, 2011, pp.22

2. *American Memory*, URL : memory.loc.gov/ammem/index.html

3. Gallica-BNF, URL : <https://gallica.bnf.fr/accueil/fr/content/accueil-fr?mode=desktop>

4. Ces exemples sont tirés de Florence Gillet, Simon Hengchen, Seth Van Hooland, Michael Sinatra et Max De Wilde, *Introduction aux humanités numériques : méthodes et pratiques*, De Boeck supérieur, 2016, pp.82

5. Marie-Françoise Limon-Bonnet, Jean-François Moufflet et Gaetano Piraino, « L'innovation numérique : un cercle vertueux pour l'archivistique », *La Gazette des archives*, 254-2 (2019), p. 247-281, URL : <https://www.archivistes.org/Les-Archives-nationales-une-refondation-pour-le-XXIe->

1.1 Lectaurep, un enfant né de la rencontre du numérique et du patrimoine

alors de fournir rapidement des données au public, aux chercheuses et aux chercheurs. Rendre le savoir accessible à tous, le mettre en commun via des plate-formes *web* et permettre une meilleure conservation des documents. On note souvent que cette « révolution de la numérisation » dans les institutions patrimoniales ne s'est pas faite sans douleur⁶. Les pratiques traditionnelles d'approche des sources des chercheurs en sciences humaines allaient être modifiées. Désormais celles-ci devenaient numériques, elles nécessitaient de développer de nouvelles compétences et de suivre des formations différentes. En effet, la numérisation des collections a engagé des horizons d'attente, tant formels qu'intellectuels.

1.1.3 L'essor des humanités numériques

En parallèle de cette mutation opérée dans les institutions patrimoniales, le milieu universitaire était bouleversé par l'**essor des humanités numériques**⁷. Le paradigme des sciences humaines et sociales évoluait au prix de l'accroissement de l'usage des outils d'ingénierie informatique et d'une modification des méthodes propres aux disciplines des SHS, parmi ces dernières l'histoire. Cependant, la prise de conscience du rôle de l'ordonnateur dans l'historiographie ne date pas de la naissance des Humanités numériques.

En 1968 dans un article intitulé « La fin des érudits » paru dans *Le Nouvel Observateur*, l'historien Emmanuel Le Roy Ladurie concluait que « L'historien de demain sera programmeur ou ne sera pas ». En 1977, les Actes du colloque de Rome des 20-22 mai 1975 paraissaient sous le titre « Informatique et histoire médiévale »⁸, évoquant les problèmes de méthodes et les techniques informatiques pour traiter les grandes séries de documents historiques. S'appuyant sur la dématérialisation du savoir opéré depuis les années 1990 par les institutions patrimoniales et la naissance du *web*, les humanités numériques ont envisagé des visualisations et des analyses de documents basées sur la constitution de bases de données, l'analyse en réseaux, l'analyse de texte (*text mining*) et la fouille de données (*data mining*) par le biais de modèles mathématiques (probabilistes et statistiques).

siecle, pp.248

6. Parmi les difficultés rencontrées : l'aspect qualitatif est négligé au profit du quantitatif, manque de formation professionnelle, système de gestion inadapté, redondance et perte des données, faiblesse des métadonnées, phénomène de désertion des salles de lecture... De plus certains projets de numérisation privés, ce sont heurtés aux politiques nationales et internationales du droit d'auteur et de la commercialisation des œuvres via leur numérisation massive ainsi qu'au renvoi vers des sites marchands, comme cela a été le cas lors de l'affaire Google Books en 2005 ; pour plus de détails sur ce fait : Jean-Noël Jeanneney, *Quand Google défie l'Europe*, Fayard/Mille et une nuits, 2010

7. L'expression *Digital Humanities* est popularisée par l'ouvrage *A Companion to Digital Humanities* (2004) ; on considère, généralement, l'index informatisé de l'œuvre de Thomas d'Aquin réalisé par le jésuite italien Roberto Busa, en collaboration avec IBM, en 1949, comme une réalisation pionnière dans le domaine.

8. Lucie Fossier, André Vauchez et Cinzio Violante, « Informatique et histoire médiévale. Actes du colloque de Rome (20-22 mai 1975) », *Publications de l'École française de Rome*, 31-1 (1977), URL : https://www.persee.fr/issue/efr_0000-0000_1977_act_31_1 (visité le 14/09/2020)

Ces outils permettent alors de promouvoir de vastes projets de valorisation des archives⁹, mais aussi la constitution d'éditions numériques en ligne via le standard d'encodage *Text Encoding Initiative* (TEI)¹⁰. De plus, la création de blogs de recherche et de sites *web* spécialisés est facilité grâce à l'utilisation de CMS¹¹. Le récent colloque ayant eu lieu les 17 et 18 octobre 2019 au Centre des archives diplomatiques de la Courneuve, intitulé « Les archives au défi du numérique » fait état des multiples et des actuels défis posés par la collaboration entre les humanités numériques et les institutions patrimoniales¹².

1.1.4 Les potentialités des technologies OCR/HTR pour les institutions patrimoniales et leurs publics

Face à ce double mouvement de numérisation et d'essor du champ des humanités numériques, fournir un accès distant aux documents ne suffisait plus. En effet, cela ne rendait pas nécessairement ces derniers exploitables pour le public, les chercheuses et les chercheurs. L'idée est alors de pouvoir chercher une information très spécifique et localisée dans le document. Cela implique de les rendre directement requêtables grâce à une interface dédiée. C'est à ce moment que se révèle tout l'intérêt porté aux technologies de transcription automatique.

Depuis les années 1960 et jusqu'aux années 1990, les technologies de reconnaissance de caractères imprimés (*Optical Character Recognition - OCR*) et manuscrits (*Handwritten Transcription Recognition - HTR*) ont progressé jusqu'à atteindre de très bons résultats (Cf. Section 2.2). Elles étaient notamment utilisées dans des secteurs tels que les banques et les assurances. Les institutions culturelles quant à elle, ne commencèrent à réellement expérimenter ces technologies qu'à partir des années 2000.

En 2010, c'est l'*University College* de Londres (UCL) qui propose, dans le cadre du consortium européen READ (*Recognition and Enrichment of Archival Documents*) un projet d'HTR intitulé *Transcribe Bentham*¹³. Celui-ci a pour objectif de transcrire une masse considérable d'archives manuscrites, léguées par le philosophe anglais Jeremy Bentham (1748-1832). Si la BNF produisait déjà de l'OCR sur certains ouvrages imprimés du dépôt légal au travers de son interface Gallica, l'institution lance en 2012 la plate-forme

9. Voir le projet européen de l'École Polytechnique Fédérale de Lausanne et de l'Université Ca' Foscari de Venise nommé *Venice Time Machine*, URL : <https://www.timemachine.eu/>

10. Pour plus de précision Cf. Partie II

11. **Content Management System** ou CMS est une plate-forme pour déployer des sites *web*, pour gérer la mise en ligne de contenus et l'apparence visuelle, sans passer par du codage informatique. Il existe plusieurs solutions CMS comme Omeka et Wordpress, entre autres.

12. Ce mémoire n'a pas la prétention de résumer l'histoire des humanités numériques, ni même d'en faire l'analyse détaillée, cependant on se rapportera à sa Bibliographie F pour plus de précision sur le sujet.

13. *Transcribe Bentham Project*, URL : <https://blogs.ucl.ac.uk/transcribe-bentham/>

1.1 Lectaurep, un enfant né de la rencontre du numérique et du patrimoine

collaborative de recherche et de transcription (plate-forme de *crowdsourcing*) Correct¹⁴, afin de permettre aux usagers de Gallica de corriger les résultats des traitements OCR et de contrôler la qualité des corrections. L'usager devient alors acteur de la production des données qui serviront lors des phases d'entraînement des modèles de transcription.

La même année, le Centre d'études supérieures de la Renaissance (CESR) du CNRS et le laboratoire d'informatique de l'Université de Tours, présentaient un projet d'identification des caractères typographiques du XVI^e siècle¹⁵. Celui-ci a permis de montrer, grâce à l'OCR/HTR, la possibilité de proposer des contenus didactiques de « paléographie numérique ».

L'adoption par un nombre croissant d'institutions patrimoniales du protocole IIIF¹⁶ (*International Image Interoperability Framework*), a ouvert la possibilité de traiter de grandes collections d'images sur des plate-formes de stockage en ligne (*cloud*), grâce à un accès standardisé aux images de leurs collections sur le *web* et de fournir davantage de données aux logiciels HTR¹⁷.

Aux AN, c'est le projet « HIMANIS »¹⁸ (*Historical MANuscript Indexing for user-controlled Search*) qui permit un apport stratégique conséquent dans la mise en place du projet Lectaurep. « HIMANIS » est un projet de recherche européen, associant, sous le pilotage de l'IRHT (CNRS, France), la société de reconnaissance en d'écriture manuscrite A2iA (France) aujourd'hui acquise par Mitek (le projet sera par la suite repris par la société Teklia), l'Université de Groningen (Pays-Bas) et l'Université polytechnique de Valence (Espagne). Les enjeux du projet reposent alors sur la lecture automatique des registres de la chancellerie royale des XIV^e et XV^e siècles (cotes JJ35 à JJ211). L'utilisation

14. <https://www.bnf.fr/fr/plate-forme-correct>

15. Frédéric Rayar, Jean-Yves Ramel et Rémi Jimenes, *Exploiting Document Image Analysis in the Humanities*, 2012, URL : <https://halshs.archives-ouvertes.fr/halshs-00805863> (visité le 14/09/2020)

16. **IIIF** (*International Image Interoperability Framework*) est une communauté et un ensemble de spécifications techniques visant à fournir un cadre d'intégration dans la diffusion et l'échange des images en haute résolution sur le *web*. L'API Image permet de définir un service *web* pour renvoyer une image via une requête HTTP construite à partir d'une URI. Cette URI peut spécifier la région, la taille, la rotation, les caractéristiques de qualité et le format de l'image. Elles sont construites selon les besoins des applications clientes. L'API Présentation est un service *web* qui renvoie des documents structurés en JSON-LD (*JavaScript Object Notation - Linked Data*) qui donnent des informations sur la structure et la présentation de l'objet numérisé ou de la collection d'images. En somme, ce sont toutes les métadonnées accompagnant l'image qui peuvent provenir d'autres fichiers de données. Pour plus de précisions : <https://iiif.io>

17. Emanuela Boros, Alexis Toumi, Erwan Rouchet, Bastien Abadie, Dominique Stutzmann et Christopher Kermorvant, *Automatic Page Classification in a Large Collection of Manuscripts Based on the International Image Interoperability Framework*, International Conference on Document Analysis and Recognition, 2019, URL : <https://www.computer.org/csdl/proceedings-article/icdar/2019/301400a756/1h81wiF3AA0> (visité le 14/09/2020)

18. Sébastien Hamel, J.F. Moufflet et D. Stutzmann, « La recherche en plein texte dans les sources manuscrites médiévales : enjeux et perspectives du projet HIMANIS pour l'édition électronique », *Médiévales. Langues, Textes, Histoire*, 73–73 (2017), p. 67-96, URL : <http://journals.openedition.org/miedievales/8198> (visité le 16/08/2020)

du logiciel *Transkribus*¹⁹ pour le développement du modèle HTR, couplé à un moteur de recherche spécifique permet actuellement la recherche en texte intégral d'informations dans plus de 68 000 chartes du Trésor des Chartes²⁰.

1.1.5 La mise en place d'un projet HTR au département du minutier central des Archives nationales

La mise en place d'un projet HTR, au département du Minutier central des notaires de Paris, aux AN, n'a pas été aussi évidente. Michel Ollion, ancien adjoint au chef du département du Minutier central, aujourd'hui conservateur des registres de la chancellerie, propose en 2005²¹ l'expérimentation d'un accès automatique au contenu d'un répertoire par l'intermédiaire d'un logiciel de reconnaissance HTR développé par l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) de Rennes et qui a connu un succès sur les registres matricules militaires du XIX^e siècle numérisés des Archives départementales des Yvelines.

Appliqué aux répertoires de notaires, ce logiciel devait permettre une vaste opération d'indexation des personnes et des biens dans les images constituant une évolution par rapport à la consultation traditionnelle sur microfilms. Le projet ne donne alors pas de suite directe.

En juin 2016, lors du renouvellement du Projet scientifique culturel et éducatif (PSCE) triennal des AN, Marie-Françoise Limon-Bonnet, conservatrice en chef du DMC, et Aurélia Rostaing, directrice du pôle instruments de recherche, proposent de relancer le projet d'HTR sur les répertoires. L'objectif est de se concentrer sur les répertoires de contrats de mariage des négociants. Ce corpus, de petite taille et présentant une graphie stable se prêtait tout à fait à cet exercice.

Cependant, le manque de partenaires et l'attente des conclusions d'« HIMANIS », mettent le projet en attente. Les résultats prometteurs du projet « HIMANIS » arrivant peu de temps après ainsi que la recherche d'un partenariat entre le Ministère de la Culture et INRIA en 2016, pour lancer des projets innovants, marquent le commencement du projet Lectaurep pour le DMC.

19. Transkribus est un logiciel avec une interface graphique (GUI) développé en 2013 dans le cadre du projet READ (*Recognition and Enrichment of Archival Documents*). Il permet de segmenter une page, de la transcrire (manuellement ou automatiquement) et de l'annoter avant de l'exporter dans plusieurs types de formats. Le moteur HTR a été développé par le *Computational Intelligence Technology Lab* (CITlab) de l'université de Rostock.

20. « HIMANIS », URL : <https://www.himanis.org/>

21. Cf. Annexes A,
/A-Sources_et_Ecosystème_Lectaurep//A1-histoire_projet_lectaurep/demande_ocr_Ollion.pdf

1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »

1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »

1.2.1 Le département du minutier central et les répertoires de notaires

Le département du Minutier central des notaires de Paris des Archives nationales conserve les archives notariales. Ce département est créé par la loi du 14 mars 1928²² sous l'impulsion de l'archiviste Ernest Coyecque (1864-1954). Aux termes de l'article L.213-2 du Code du Patrimoine, les archives notariales sont communicables aux public 75 ans après leur date de production, à compter de la date de création de l'acte. À noter que cette échéance s'abaisse à 25 ans après le décès des parties concernées ou s'élève à 100 ans si l'acte concerne une personne mineure²³.

Le DMC assure des missions de traitement archivistiques : collecte, conservation, signalement dans les instruments de recherche, communication, et valorisation des archives notariales.²⁴

On estime que les fonds du DMC s'élèvent à 20 millions d'actes notariés. Soit 20 km linéaires, ce qui est en fait le « plus grand minutier du monde ». Ces actes provenant des CXXII²⁵ études de notaires qui constituent les 122 fonds composés de minutes et de répertoires de notaires²⁶.

En droit français, on définit la minute²⁷ d'un acte notarié comme l'original d'un acte authentique, dont le notaire ne peut se défaire. Les parties recevant des copies de l'acte. On parle d'actes en minute par opposition aux actes en brevet : un acte en brevet est un acte notarié dont l'original est dépourvu de formule exécutoire, et est remis aux parties.

22. Loi du 14 mars 1928 relative au dépôt facultatif dans les AN et départementales des actes datés de plus de 125 ans conservés dans les études de notaires, JORF, no 64, 15 mars 1928, p. 2830., URL : <https://gallica.bnf.fr/ark:/12148/bpt6k20282129/f2>

23. M.F. Limon-Bonnet et Geneviève Étienne, *Les archives notariales : manuel pratique et juridique*, la Documentation française, Paris, 2013

24. M.F. Limon-Bonnet, J.F. Moufflet et G. Piraino, « L'innovation numérique : un cercle vertueux pour l'archivistique »..., pp.254.

25. Notation historique

26. M.F. Limon-Bonnet, Claire Béchu, Christian Lefèvre et Agnès Magnien, *122 minutes d'histoire. Actes des notaires de Paris XVIe-XXe siècle*, Somogy éditions d'art, 2012, pp. 4

27. Pour voir un exemple de minute, Cf. Annexes, Figure A.1

Le terme « minute », tirerait son origine du latin médiéval *minuta* signifiant « partie menue », que l'on peut traduire par « résumé », « note » ou « brouillon »²⁸. De plus elle était rédigée en écriture fine, pour des raisons d'archivage, par opposition à la « grosse » ou « expéditions », qui sont les copies délivrées aux parties. Ces minutes constituent des sources inestimables pour les historiens. Parmi celles-ci on peut citer des contrats de mariage (Mariage de Jacques Offenbach, 1844, MC/ET/XII/717) et de divorce (Séparation des corps des époux Beauharnais, 1785, MC/ET/LVIII/531), des testaments (Testament de Blaise Pascal, 1662, MC/ET/XVII/32), ou encore des actes de société (création de la société de la *Revue des Deux Mondes*, 1845, MC/ET/XXXIII/1166/B)²⁹.

Les répertoires (Figure 1.1) sont des registres où les notaires consignaient tous les actes conservés en minutes. Les répertoires sont donc utilisés comme des instruments de recherche par les chercheuses, les chercheurs et les archivistes, pour repérer les minutes dans les fonds.

Les cotes de répertoires de notaires suivent la logique suivante :

- **MC** pour minutier central ;
- **RE** pour indiquer qu'il s'agit d'un répertoire de notaire ;
- le **numéro de l'étude** mentionné en chiffres romains ;
- le **numéro du répertoire** en chiffres arabes.

28. « Minute », CNRTL (Centre national des ressources textuelles et lexicales), URL : <https://www.cnrtl.fr/definition/academie9/minute//1>

29. Pour plus d'exemples de minutes voir *Ibid.*

1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »

L'ancienne famille
2

N° des ACTES	DATES	NATURE ET ESPÈCE DES ACTES :	INDICATIONS, SITUATIONS ET PRIX DES BIENS		RELATION DE l'Enregistrement.
			EN BREVETS	EN MINUTES	
43	8	mandat			
44	8	○ mariage			9 7.38
45	9	Bail			10 28.13
46	9	Dépot des ornaments de la reine			12 109.50
47	9	Procuration			9 11.25
48	9	Discharge			10 3.75
49	9	Procuration			10 3.75
50	9	Procuration			10 3.75
51	9	Testament			10 3.75
52	9	Procuration			10 3.75
53	9	Substitution			10 3.75
54	10	Vent de f ^{me} de com.			12 3.75
55	10	Tenure			15 2409.80
56	10	Concédé Bail			14 18.75
57	10	25 sept 1919, substitution			13 8.75
58	10	Substitution de Simon Obligation			13 3.75

FIGURE 1.1 – Exemple de répertoire de notaire ©Archives nationales/DMC, MC/RE/X-LIII/42, étude XLIII du notaire Louis Marie Joseph Marotte

Les répertoires de notaires ont fait l'objet de vastes campagnes de numérisation, sous la forme de double-pages par la société Arkanum³⁰. Ces images sont, pour la plupart, accessibles en ligne dans la Salle des inventaires virtuelles des AN.

La structure régulière des répertoires (structure homogène en tableau rendue obligatoire et formalisée par les articles 29 et 30 de la loi du 25 Ventôse an XI) et les exploitations scientifiques potentielles, font de ces documents des candidats idéals pour un projet de reconnaissance. Dès lors, le choix du corpus pour le projet s'est constitué autour d'environ deux mille répertoires de neuf cents notaires, pour la période allant de 1803 à 1944. Chacun de ces répertoires comprenant entre trois cents et cinq cents pages.

1.2.2 Le projet Lectaurep : cadre, avancées et objectifs de la phase 3

Les objectifs du projet Lectaurep sont à terme de proposer³¹ :

1. Un outil de recherche intégral pour les publics des archives ;
2. Des fonctionnalités d'analyses statistiques pour les chercheurs, s'appuyant sur les humanités numériques et les outils du Traitement automatique du langage (TAL), sur lesquels nous reviendrons, plus en détail, dans la section 2.3 ;
3. Récupérer des informations des traitements HTR d'*eScriptorium* dans la SIV ;
4. Un outil en réseau, mutualisé (*crowdsourcing*) pour les services d'archives, pour forger une communauté autour de la transcription des répertoires.

Pour répondre à ses besoins, Lectaurep a envisagé la technologie de reconnaissance automatique de structure et d'écriture manuscrite (HTR) ainsi que l'indexation. Enfin, les résultats seront publiés sur une plate-forme dotée d'un moteur de recherche avancée, permettant de requêter directement dans les images de répertoires.

Le projet (convention-cadre signée entre INRIA et le Ministère de la Culture en 2016) a connu plusieurs étapes essentielles. Une première phase exploratoire (débutée en 2018), a permis de brosser un état de l'art de la reconnaissance automatique et les avantages que celle-ci offre au projet Lectaurep. Ainsi un premier travail de repérage des tableaux³² dans les répertoires de notaires a été mené par Marie-Laurence Bonhomme, alors stagiaire du

30. Marie-Laurence Bonhomme, *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris*, Mémoire de recherche, École nationale des chartes, 2018, pp. 16

31. Alix Chagué, *LECTAUREP Lecture Automatique de Répertoires*, Atelier Culture, 2019, URL : <https://webcache.googleusercontent.com/search?q=cache:SS8LEFv8NJIJ:https://www.culture.gouv.fr/Media/Thematiques/Innovation-numerique/Folder/Atelier-Inria-2018/Lectaurep-lecture-automatique-de-repertoires+&cd=1&hl=fr&ct=clnk&gl=fr&client=firefox-b-d>

32. Pour avoir une idée du repérage des tableaux Cf. Annexes, Figure A.2

1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »

master TNAH de l'École nationale des chartes. Dans son rapport exploratoire³³, elle émet les premières préconisations quant au travail de segmentation des tableaux avec le logiciel *Transkribus*. Cependant, elle constate, une mauvaise prise en charge des documents de structure tabulaire par le logiciel et montre les très bons résultats du logiciel *Kraken*, tout juste développé : il ne possédait pas encore d'interface graphique et la segmentation n'était pas encore tout à fait au point.

La phase 2 (2019) du projet a connu plusieurs évolutions, en cause les réorientations dans les choix d'outils, dans l'acquisition des données et la gestion de projet :

- Constitution de jeux de données (images de répertoires de notaires) mis à disposition de l'équipe ALMAaCH par le Département de la maîtrise douvrage du système d'information (DMOASI) des AN sur l'espace *ShareDocs* d'Huma-Num répartis en un *Golden set* et un *Random set* ;
- Abandon de l'outil *Transkribus* et migration des données au profit des logiciels *eScriptorium* et de sa brique OCR/HTR *Kraken* ;
- Mise en place d'un espace collaboratif basé sur *GitLab* pour partager les scripts nécessaires en appui à la chaîne de traitement Lectaurep (*Pipeline* entre Transkribus et eScriptorium pour le transfert et la compatibilité des vérités terrains comme le programme Aspyre GT³⁴ d'Alix Chagué, ingénierie en humanités numériques à ALMAaCH pour le projet Lectaurep).

L'écosystème de travail de Lectaurep, présenté dans la liste n'a, dans son ensemble, pas été modifié depuis. Les images de répertoires sont réparties en deux *sets* sur *ShareDocs*³⁵ :

1. Le *Golden set* : composé d'environ mille doubles pages de quarante-et-un registres (couvrant la période 1789-1875) numérisés en noir et blanc et en couleur. Il doit servir de base pour créer des vérités terrains ;
2. Le *Random set* : second set composé mille doubles pages aléatoires de quatre campagnes de numérisation récentes en couleur (allant des années 1880 à 1930), doit permettre de tester les modèles de segmentation et de transcription avec des données mixtes et de créer un modèle général.

33. M.L. Bonhomme, *Répertoire des Notaires parisiens Segmentation automatique et reconnaissance d'écriture*, rapport, Inria, 2018, URL : <https://hal.inria.fr/hal-01949198> (visité le 13/09/2020)

34. Alic Chagué, Aspyre GT, *A pipeline to transfer ground truth from Transkribus to eScriptorium*, URL : <https://gitlab.inria.fr/dh-projects/aspyre-gt>

35. Cf. Annexes, Figure A.4

L'abandon de l'outil *Transkribus* au profit du couple *Kraken-eScriptorium*, correspond au besoin pour Lectaurep d'avoir la main sur le système d'entraînement des modèles de transcription et de segmentation. En effet si l'interface de *Transkribus* est en accès libre, en revanche les modèles de transcription et les résultats ne sont accessibles que par l'intermédiaire de l'équipe dudit logiciel, qui deviendra payant à terme.

*Kraken*³⁶ est le principal logiciel de reconnaissance en ligne de commande (CLI - *Command Line Interface*) utilisé dans le cadre du projet Lectaurep. Il a été développé sur la base du logiciel *OCRopus* par Benjamin Kiessling. *Open-source*, il permet de binariser les images, de segmenter et d'entraîner des modèles d'OCR/HTR. Basé sur les réseaux de neurones, il permet d'obtenir de très bons résultats, aussi bien sur les documents imprimés que sur les documents manuscrits dans des caractères latins, hébreux et arabes, avec des taux d'erreur parfois inférieurs à 2%³⁷. Il est également doté d'une API^{38 39} (*Application Programming Interface*) qui permet de récupérer le code source pour réutiliser des fonctionnalités spécifiques dans d'autres projets (nous verrons un cas d'application de cette API dans la Partie III).

*eScriptorium*⁴⁰ est l'interface graphique dont s'est doté *Kraken* dans le cadre du projet « Scripta » de l'Université PSL⁴¹. Il s'agit de l'interface *web* utilisée par les annotatrices et les annotateurs de Lectaurep aux AN, pour réaliser la segmentation et transcrire les images afin de préparer les données d'entraînement (vérités terrains).

La plate-forme *eScriptorium* est encore en cours de développement. Dès lors, ALMAnaCH est très souvent confronté aux remontées de *bugs* (erreurs d'affichage, mauvais tracés des lignes de segmentation, export de la transcription faussée etc.). L'absence d'un développeur dédié à ALMAnaCH (en cours de recrutement) pour les mises à jour et les corrections de *bugs* sur eScriptorium peut ralentir la chaîne de traitement du côté des AN et l'entraînement des modèles de segmentation, ainsi que les tests du côté ALMAnaCH.

36. Kraken, a turn-key OCR system optimized for historical and non-Latin script material, URL : <http://kraken.re/>, URL du code source : <https://github.com/mittagessen/kraken>

37. Benjamin Kiessling, Sarah Bowen Savant, Maxim Romanov et Matthew Thomas Miller, « Important New Developments in Arabographic Optical Character Recognition (OCR) », *CoRR*, abs/1703.09550 (2017), URL : https://www.academia.edu/28923960/Important_New_Developments_in_Arabographic_Optical_Character_Recognition_OCR_ (visité le 13/09/2020)

38. **API** (*Application Programming Interface*) ou interface de programmation d'application permet de réutiliser du code déjà écrit dans d'autres applications pour le réutiliser dans son propre programme. Généralement documentée par le service, il s'agit d'un ensemble de fonctions, classes, méthodes et variables réutilisables dans son propre code.

39. *Kraken API*, URL : <http://kraken.re/api.html>

40. Cf. Annexes, Figure A.3

41. Scripta PSL, *Scripta PSL / Histoire et pratiques de l'écrit, PSL Scripta*, URL : <https://scripta.ps1.eu/> (visité le 13/09/2020)

1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »

La phase 3, durant laquelle le stage s'est déroulé, poursuit ces efforts dans l'obtention de meilleures données d'entraînement pour réaliser des modèles de segmentation plus performants. Cela grâce, notamment à des paramétrages différents de *Kraken* pour l'entraînement des modèles. Les AN poursuivent leur travail d'annotations sur *eScriptorium* pour obtenir davantage de données d'entraînement. De plus, un certains nombres de fonctionnalités sont en cours de développement pour *eScriptorium* du côté de l'équipe de « Scripta ». ALMAnaCH développe également en interne plusieurs fonctionnalités pour Lectaurep hors *eScriptorium*. L'ensemble de ces fonctionnalités en cours de développement est résumé dans la Figure 1.2. Le stage était axé sur le développement de deux de ces fonctionnalités hors *eScriptorium* à savoir : la génération d'un fichier XML-TEI pivot pour les métadonnées (Cf. Partie II) et la création d'un outil interne pour préparer l'évaluation des modèles de transcription (Cf. Partie III).

De plus afin d'améliorer la visibilité du projet pour permettre un retour d'expérience de la part des acteurs de Lectaurep, pour servir à d'autres projets similaires, et centraliser la documentation et l'histoire du projet, nous avons mis en place un blog *hypothèses* Lectaurep⁴². Basé sur le CMS *Wordpress*, j'ai organisé le site *web* et préparé l'environnement de rédaction associé pour accueillir de futurs articles qui sont d'ores et déjà en cours de rédaction.

42. Cf. Annexes A, Figure A.5, URL du blog : <https://lectaurep.hypotheses.org/>

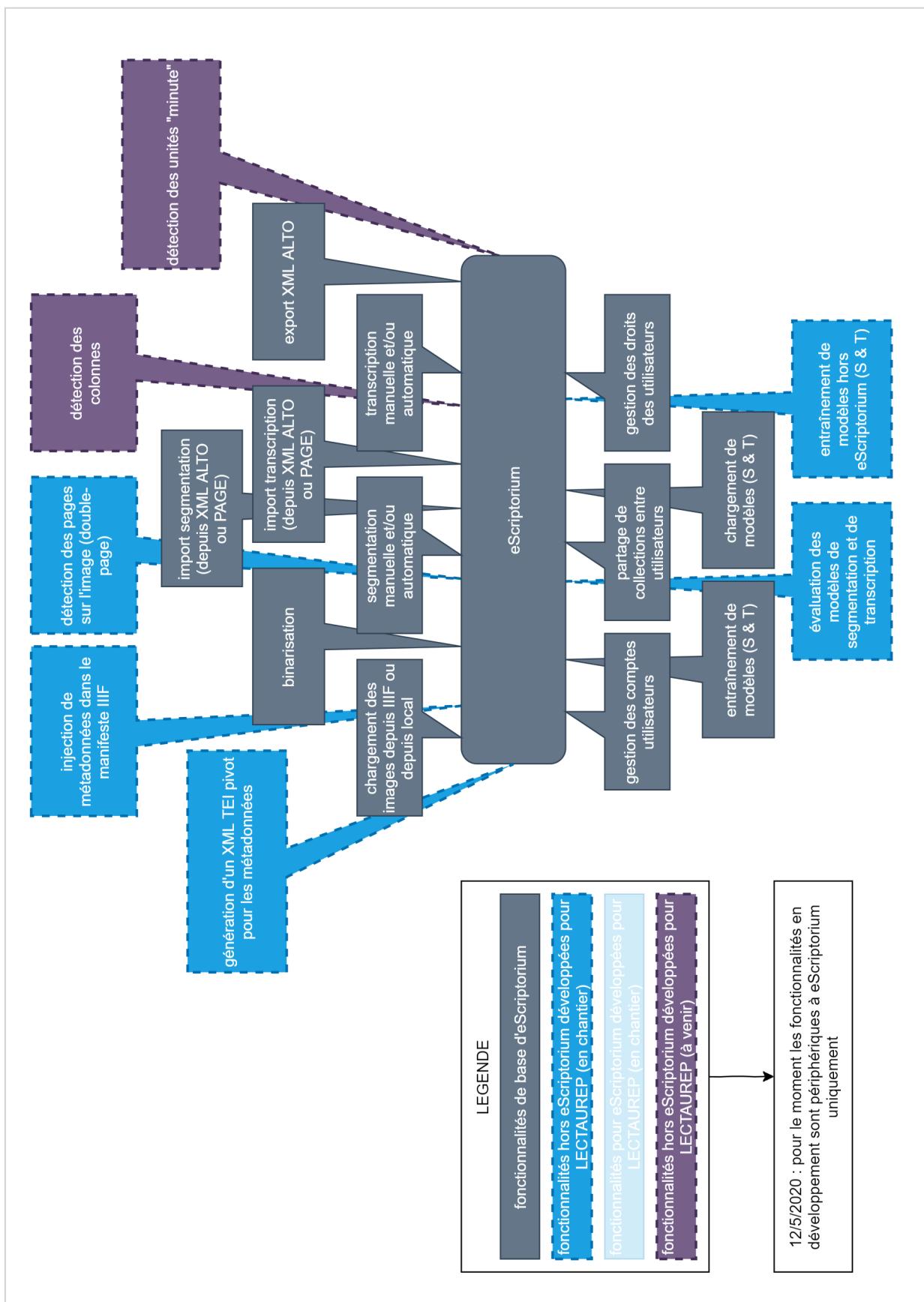


FIGURE 1.2 – Illustration des fonctionnalités en cours de développement durant la phase 3 de Lectaurep ©A. Chagué, 2020, Diagrams.net

Chapitre 2

La reconnaissance automatique des écritures dans Lectaurep : un domaine de l'intelligence artificielle et du traitement automatique du langage naturel

2.1 Définir les composantes de l'intelligence artificielle dans le projet

2.1.1 Les champs de l'intelligence artificielle

En mars 2018, le mathématicien Cédric Villani rend public le rapport, issu d'une mission parlementaire, intitulé « Donner un sens à l'intelligence artificielle : pour une stratégie nationale et européenne »¹, dans lequel il mène une réflexion détaillée sur l'état de l'art et les atouts de l'intelligence artificielle (IA) en France. Dans ce rapport il apparaît que la France compte parmi les quatre premiers au monde, avec la Chine, les États-Unis et le Royaume-Uni pour la production mondiale d'articles sur l'IA et rend compte d'une définition de l'IA non comme :

1. Cédric Villani, *Rapport de Cédric Villani : donner un sens à l'intelligence artificielle (IA)*, Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation, 2018, URL : //www.enseignementsup-recherche.gouv.fr/cid128577/rapport-de-cedric-villani-donner-un-sens-a-l-intelligence-artificielle-ia.html (visité le 11/09/2020)

Un champ de recherches bien défini qu'un programme, fondé autour d'un objectif ambitieux : comprendre comment fonctionne la cognition humaine et la reproduire ; créer des processus cognitifs comparables à ceux de l'être humain.²

Cette définition n'est peut-être pas la plus complète ni la seule qui existe³, mais elle possède l'avantage de définir l'IA comme un champ de recherche en informatique théorique et comme la création de systèmes qui imitent les performances humaines. En 1950, Alan Turing proposait un test hypothétique⁴ afin de savoir si un ordinateur avait acquis l'intelligence opérationnelle d'un humain.

Le principe était le suivant : après une série de questions posées à l'ordinateur par un humain, le test était réussi si l'humain en question était dans l'incapacité de dire si les réponses provenaient d'un autre humain ou d'un système informatisé.

Dès lors, pour passer ce test, et se confondre à l'humain, l'ordinateur devrait posséder les fonctionnalités suivantes : **le traitement du langage naturel**, pour communiquer, la **représentation des connaissances**, sous la forme d'une mémoire, un **raisonnement automatisé**, pour tirer des conclusions logiques de l'expérience mémorisée et l'**apprentissage**, pour ajuster ses réponses aux circonstances auxquelles il se retrouve confronté et s'adapter au hasard. Enfin pour simuler entièrement l'humain et passer le test de Turing dit « complet », la perception du système pourrait être vérifiée à l'aune de la **vision artificielle**, pour percevoir les objets, et la **robotique** pour les manipuler⁵.

Parmi les applications de l'IA les plus connues, les véhicules autonomes à l'image de la voiture robotisée de l'université de Stanford en 2005, la reconnaissance de la parole, avec les assistants personnels comme *Alexa* (2014), les jeux tel que *Deep Blue* d'IBM, le super ordinateur qui a battu le champion mondial Garry Kasparov aux échecs en 1997 suivi par *Alpha Go* de Google Deepmind en 2015 qui bat le champion du monde du jeu de go. On peut également penser aux systèmes de traduction automatique, de type *Google Translate*, qui voit le jour en 2006. Plus récemment la propagation du virus Covid-19, a accentué les usages de l'IA dans les domaines de la santé, notamment pour identifier certains foyers (*clusters*) sanitaires localisés avec plus ou moins de succès en se basant sur les données médicales.

2. *Ibid.*, pp.9

3. Stuart Russell et Peter Norvig, *Intelligence artificielle : Avec plus de 500 exercices*, Pearson Education France, 2010, pp.4

4. Alan M. Turing, « Computing Machinery and Intelligence », *Mind*, LIX–236 (1950), p. 433-460, URL : <https://academic.oup.com/mind/article/LIX/236/433/986238> (visité le 11/09/2020)

5. S. Russell et P. Norvig, *Intelligence artificielle : Avec plus de 500 exercices...*, pp. 3

2.1 Définir les composantes de l'intelligence artificielle dans le projet

La plupart de ces applications sont développées dans le cadre d'une « IA faible », c'est-à-dire des programmes basés sur des algorithmes⁶ capables de réaliser une tâche précise pour laquelle ils ont été entraînés. On parle alors d'« IA forte » pour qualifier tout système qui s'affranchirait des volontés humaines et développerait une « singularité technologique »⁷. Cependant ces systèmes pouvant réaliser des tâches en parfaite autonomie, sans contrôle humain, appartiennent encore au domaine de la science-fiction (à l'image des robots de l'écrivain Isaac Asimov (1920-1992) qui sont soumis aux trois lois de la robotique pour protéger l'humanité des dérives potentiels de ces derniers).

D'après ces exemples et le rapport Villani évoqué plus haut, les secteurs prioritaires de l'IA concernent avant tout : la santé, les transports, l'environnement et la défense.

Comment identifier l'actuelle plus-value de l'IA appliquée à la culture et aux projets patrimoniaux ? Comment le projet Lectaurep illustre-t-il ce mouvement des institutions patrimoniales vers l'IA ? La reconnaissance des écritures manuscrites implique de visualiser une image et de détecter le texte : ce qui suppose de disposer de méthodes de perception visuelle, suivre le tracé de l'écriture puis reconnaître les caractères (grâce à des algorithmes de reconnaissance de formes) et enfin reconnaître les mots et les phrases par le traitement automatique de la langue pour aller jusqu'à les comprendre (via une modélisation sémantique). Les systèmes de reconnaissance de l'écriture manuscrite utilisés dans Lectaurep, en prenant appui sur le développement des réseaux de neurones, concernent l'IA et plus précisément le domaine du *deep learning* (DL).

Le DL est une sous discipline du *machine learning* (ML), qui est elle-même une sous-branche de l'IA (Cf Figure 2.1). Le ML et le DL diffèrent dans la manière qu'ils ont de gérer les données présentées en entrée et dans les technologies utilisées.

Dans un projet ML, le ou la *data scientist* (personne chargée des projets impliquant la gestion de données) choisit des données selon des critères définis (*feature extraction*) à l'avance pour y appliquer des modèles mathématiques de prédiction statistiques (comme le modèle de la régression linéaire, par exemple). Dans l'optique, par exemple, de réa-

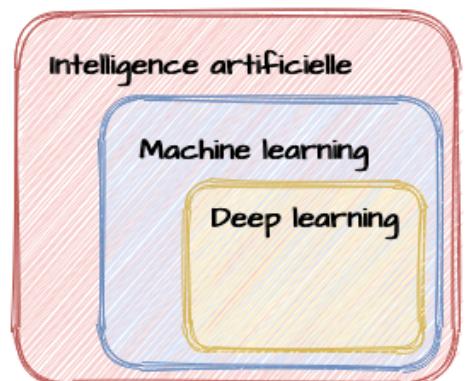


FIGURE 2.1 – Les différentes disciplines de l'IA ©L. Terriel, 2020, Diagrams.net

6. On définit un **algorithme** comme une suite d'instructions qui permet d'aboutir à un résultat donné. Une recette de cuisine ou un ensemble de directives pour aller d'un endroit A à un endroit B est un algorithme. En Informatique, il s'agit d'une séquence d'étapes implémentée dans un langage (code) permettant de réaliser un programme ou une tâche précise de ce dernier.

7. Gabriel Ganascia, *Le Mythe de la Singularité*, Seuil, 2017

liser des prédictions d'achats ; des données comme l'âge, le sexe, le revenu, les goûts musicaux, les sports préférés ou autre, pourront être exploitées.

La méthode du DL diffère en ce sens qu'il n'y a pas de *feature extraction* réalisée en amont du projet. Dès lors les données présentées en entrée sont dites « non structurées » : images, textes, sons, etc. En utilisant des algorithmes de réseaux de neurones profonds, on charge le programme d'extraire par lui-même les caractéristiques des données d'exemple en entrée et d'émettre des prédictions à partir de modèles statistiques et mathématiques, empruntés au ML. Ainsi, dans un projet DL de reconnaissance de formes basé sur des images, le programme sera capable de relever des amas de pixels caractéristiques et redondants correspondants à une forme particulière. Cependant nous verrons que dans le cas d'un apprentissage semi-supervisé, comme dans Lectaurep, certains types de données peuvent être étiquetés. Par exemple les zones contenant du texte sur les différents tableaux du répertoire sont représentées sous la forme de coordonnées de polygones et de lignes de textes. Cela permet d'améliorer les prédictions de transcription grâce à un modèle entraîné par des réseaux de neurones.

Nous allons par la suite, appréhender les différentes étapes à initier lors d'un projet de reconnaissance d'écritures manuscrites tel que Lectaurep. Cela s'apparente aux traitements classiques du *deep learning*, à savoir la préparation des données, l'entraînement des modèles, et enfin prédiction par la machine.

2.1.2 L'étape de préparation et d'acquisition des données d'apprentissage de Lectaurep

À l'ère du *Big Data*, les données sont partout et constituent la base des projets DL. Par exemple un modèle devant reconnaître des motos et des voitures dans des images, devra disposer d'un grand nombre d'exemples assez variés pour, à terme, être capable d'opérer les bonnes distinctions dans un corpus d'images représentant plusieurs moyens de transports différents. Encore faut-il que ces données, pour permettre une classification correcte, soient de bonne qualité.

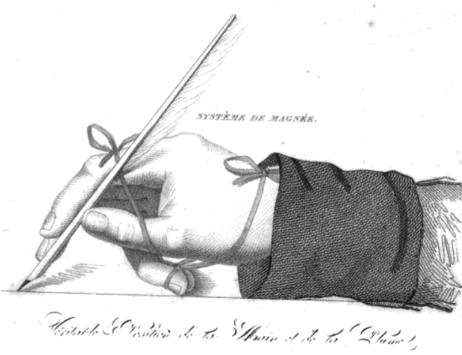
Dans le projet Lectaurep, ces données sont les images de répertoires de notaires. Ce sont elles qui vont servir de base d'entraînement au modèle. Cependant, à la différence des écritures imprimées, généralement lisibles par une machine, les écritures manuscrites constituent un problème d'un autre ordre :

2.1 Définir les composantes de l'intelligence artificielle dans le projet

Certaines écritures restent difficiles à déchiffrer [...] comme les écritures d'archives et historiques, car à la complexité d'une écriture que seuls les paléographes peuvent déchiffrer, s'ajoute la compréhension d'une langue qui n'est plus parlée ou qui a évoluée. Pourtant toutes ces écritures ont été produites par des hommes avec l'objectif de se faire comprendre sans erreur par d'autres hommes.⁸

Les écritures du XIX^e siècle contenues dans les répertoires de notaires constituent un réel défi pour la lecture machine. On constate de grandes variabilités de formes d'écriture d'un scribe à un autre. Aux aspects graphologiques liés à la pression de la plume, à l'inclinaison des traits des lettres, à l'alignement des mots par rapport aux lignes de bases, ainsi qu'aux formes des lettres qui varient d'un clerc à un autre, s'ajoutent la spécificité des outils utilisés suivant le contexte d'écriture. Pour la calligraphie, le clerc préférera la plume d'oie, tandis que pour écrire plus vite ou pour « expédier » il choisira la plume en métal sous la dictée du notaire. Cela induit le fait que, parfois dans un même répertoire de notaire, des formes de polices d'écritures très différentes peuvent se côtoyer : ronde, gothique, minuscule caroline, italique ou cursive anglaise, lettres capitales etc. Les écritures des répertoires sont rarement uniformes (Cf. Figure 2.2).

8. C. Kermorvant, *La reconnaissance d'écriture manuscrite*, Data Analytics Post, 2019, URL : <https://dataanalyticspost.com/la-reconnaissance-decriture-manuscrite-de-nouvelles-applications-pour-un-des-plus-vieux-problemes-dia/> (visité le 11/09/2020)



Deux genres
et des 2 nomb. Qui, que, dont, de
qui, à qui.

2.

TENUE DE LA PLUME. La plume sera fixée entre l'extrémité du pouce et du medium ou 3^{me} doigt. L'index ou premier doigt joindra le medium à la hauteur de l'ongle et l'ongle du pouce ne devra jamais toucher l'index. Le bout opposé au bec de la plume devra être tourné directement vis-à-vis la jointure de l'épaule droite, on ne doit jamais la faire tourner entre les doigts mais s'en servir comme d'un crayon.

Mémoires		
rédaction et la composition	Exercices	Exercices

Monsieur Dupré se promenait un jour dans les champs avec Félix, son plus jeune fils. C'était un beau jour d'automne, et il faisait encore fort chaud.



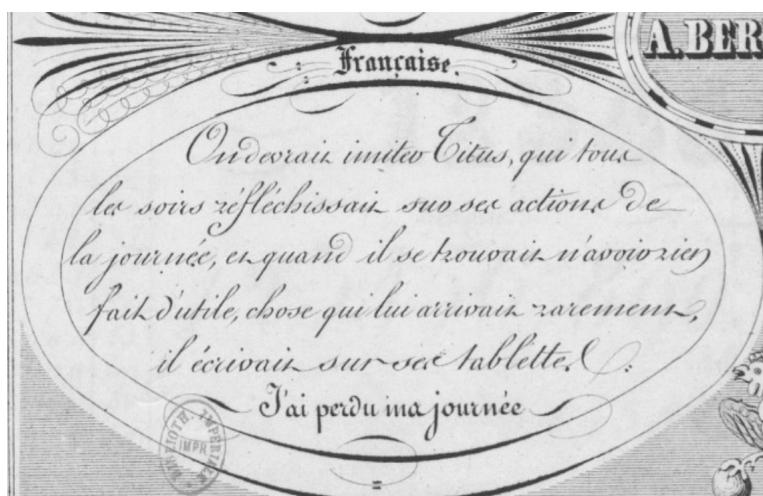
2.1 Définir les composantes de l'intelligence artificielle dans le projet

Ce sont les Hollandais qui vers l'année 1730 inventerent ce genre d'écriture et qui en ont fait usage dans leurs relations commerciales.

32

Expédiée mixte & Ronde minuscule.

Les Français semblent être maintenant les propagateurs de diverses écritures, en Europe. Les Italiens ont totalement négligé leur écriture. Les Allemands n'ont pas changé la leur. Les Hollandais suivent toujours celle adoptée par les Anglais. Il existe en France trois écritures proprement appelées françaises, ce sont la Bataille, la foulée, & la Ronde; ou trois sortes, sont encore utilisées l'Anglaise, les Gothiques & Romaines. Les français fournissent aux autres nations, du modèle, de toute cette écriture.



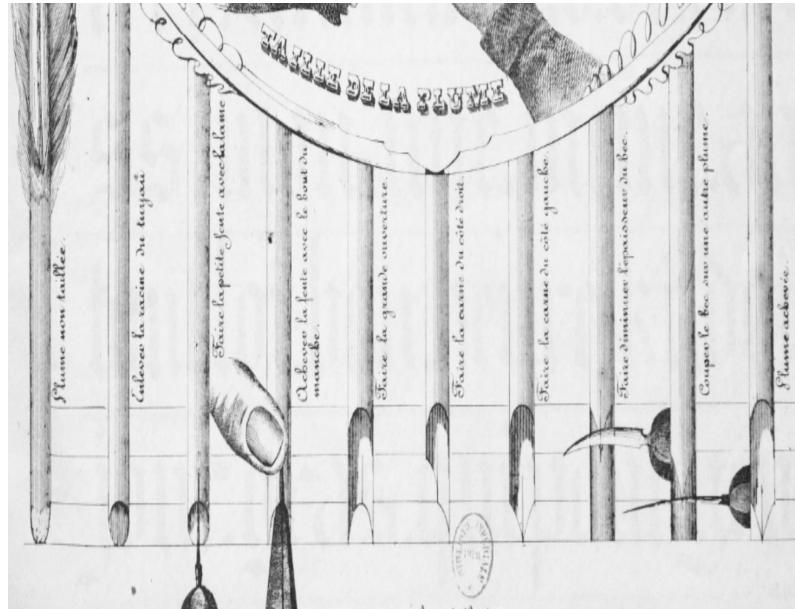


FIGURE 2.2 – Exemples de types d’écritures XIX^e siècle rencontrées dans les répertoires de notaires, de gauche à droite et de haut en bas : système d’écriture dit « de Magnée » extrait de MAGNÉE (François), *Le parfait calligraphe, ou méthode pour apprendre soi-même à écrire en peu de leçons*, 1828, exemple de « d » delta et de « Q » majuscule archaïque repris de MOLLIARD et HINARD, *Méthode pratique et simultanée de lecture, d’écriture et d’orthographe*, 1861, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k6152346g> (visité le 11/09/2020), exemples de modules de tracés des jambages et des hampes WERDET (Jean-Baptiste), *Innovation : leçons d’écriture simplifiée, par Werdet père...* 1841, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k130683n> (visité le 11/09/2020), d’écriture « coulée » Ancien Régime repris de FRÉMONT (E.-L.), *Cahiers manuscrits, recueil de toutes sortes d’écritures lithographiées, pour exercer à la lecture des écritures difficiles*, 1837, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k1162328s> (visité le 11/09/2020), exemples de gothique moderne dite « fracture », de cursive anglaise, de ronde minutée, d’écriture latine et de taille de plume extraits de BERLINER (Arnold), *Cours complet de tous les genres d’écritures usités en France, dédié à ses élèves*, 1862, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k164353h> (visité le 11/09/2020).

2.1 Définir les composantes de l'intelligence artificielle dans le projet

En plus de ces écritures très différentes, les images de répertoires n'ont pas été numérisées avec la même qualité : on retrouve parfois du noir et blanc ou des cadrages non-homogènes⁹. Dès lors ces images, avant de servir d'exemples d'entraînement, peuvent subir des **pré-traitements** comme la binarisation (passage au noir & blanc), la découpe des doubles pages des images de répertoires pour obtenir un tableau par image¹⁰, et un recadrage pour recentrer l'image.

Une fois ces pré-traitements images effectués, on réalise généralement une extraction des caractéristiques (*features*) de l'image. On identifie les différentes zones de texte, paragraphes, lignes et parfois lettres dans une phase. Il s'agit d'une opération de **segmentation**. Durant sa phase 1, Lectaurep a utilisé l'outil de segmentation inclu dans *Transkribus* pour passer ensuite à l'usage du segmenteur de *Kraken* implémenté dans la plate-forme *eScriptorium*, afin d'effectuer une segmentation des zones des tableaux repérées par Marie-Laurence Bonhomme dans les répertoires¹¹. Ces zones correspondent à la ligne de base du texte (*baseline*) et à des ensembles de points entourant du texte (*polygons*) constituant des coordonnées. Ces zones sont généralement observables dans des exports en XML ALTO ou XML PAGE¹². Si cette segmentation peut s'effectuer à la main dans *eScriptorium* ces zones constituent elles-mêmes des données d'entraînement pour créer des modèles de segmentation, qui permettent de relever ces zones de manière automatique et ainsi gagner du temps. Cependant, les annotatrices et annotateurs de la plate-forme *eScriptorium* au DMC doivent respecter un certains nombres de règles pour tracer les lignes, parmi celles-ci : un segment doit être tracé pour les prix et pas deux même si le prix est coupé par une virgule, un point ou un espace ; l'épaisseur du pinceau doit correspondre à la hauteur de la ligne ; pas d'annotation sémantique pour distinguer les zones de textes et pas de segmentation des titres de colonnes etc.

Après cette phase de segmentation, vient celle de la **transcription manuelle** contrôlée. Elle est réalisée par les annotatrices et les annotateurs des AN du projet Lectaurep et doit servir de **vérité terrain** (*ground truth*) afin d'entraîner et de paramétriser le moteur HTR disponible dans le CLI *Kraken* (hors *eScriptorium* actuellement). Ces données de vérité terrain seront comparées avec les prédictions effectuées par le système HTR pour évaluer le taux d'erreur par transcription. Cependant, la transcription « à la main » (dans *eScriptorium*) demande d'effectuer un certain nombre de normalisations et impose,

9. Pour consulter des exemples de différentes qualités numérisation, Cf. Annexes /C-Application_Kraken_Benchmark/sets_test/sets_tests_lectaurep/

10. Alix Chagué a créé un programme en Python *choppy* pour permettre cette découpe des doubles pages, URL : <https://gitlab.inria.fr/dh-projects/choppy>

11. Cf. Figure A.2, consulter également M.L. Bonhomme, *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris...*, pp. 20-26.

12. Pour plus de précisions sur ces formats consulter la Partie II

comme pour la segmentation, des règles de transcription. Ainsi, cela évite de donner aux systèmes deux interprétations d'un même caractère.

En effet, comment retranscrire avec le clavier des ligatures (fusion de deux ou trois graphèmes pour en former un nouveau), un « s » final en forme archaïque et moderne, un « s » long, un « d » en forme de delta recourbé ou en hampe archaïque, faut-il rajouter des majuscules aux noms propres lorsqu'elles sont absentes ?

Ces questions ne sont pas encore totalement formalisées par le DMC qui effectue actuellement un travail pour clarifier ces règles de transcription. Cela reste encore une tâche difficile à réaliser tant les cas particuliers abondent (Cf. Figure 2.3).

Les étapes de préparation des données sont donc essentielles car elles déterminent les résultats. Certaines erreurs peuvent constituer des **biais de prédictions** qui faussent l'issue du traitement. Une fois les données acquises il est possible de passer à l'apprentissage pour la création des modèles.

2.1 Définir les composantes de l'intelligence artificielle dans le projet

(par In Bte) rue
à lui donnée par
so.

17. m

d'Espagne, natif de Cluny Fontenay
Province de Haute Savoie, adt. département
du Montblanc.

28

t (par In Bte) rue

Chartené (d'Elisabeth Renée) veuve de
Juan Manuel, née le 10. 7^{bre} 1786. demeurant
à Paris rue de l'arcade n° 28

province de Haute Savoie, adt. département

contenant en
900 = payés

Juan Manuel, née le 10. 7^{bre} 1786. demeurant

Raymond Clémery (bénéfice Joseph...
Montesquieu fezensac et Mme Henriette Clarke
de Bellière d'Annebourg son épouse, père & mère

900+ payés

au Palais de Justice à Paris,
le Douze Avril mil huit cent trente
ans. /

Montesquieu fezensac et Mme Henriette Clarke

un XXX ~~~~~

FIGURE 2.3 – Quelques exemples de transcriptions réalisées par des annotateurs de Lectaurep dans eScriptorium. ©Captures fournies par A. Rostaing (AN/DMC), 2020, eScriptorium

2.1.3 Apprentissage et entraînement des modèles

On distingue généralement trois types d'apprentissages machine :

- **l'apprentissage supervisé** : les données sont catégorisées par des étiquettes (*features*) selon des caractéristiques en entrée de l'algorithme. C'est une étape de préparation lourde en ressources humaines car elle est « guidé » par l'humain. L'algorithme se charge alors d'ajuster les marges d'erreurs au fil des itérations durant l'entraînement, jusqu'à créer un modèle généralisable à des données, cette fois, non étiquetées, et qui obtiennent de bons résultats. Par exemple, dans le cadre d'une application de détection de *spam*, les caractéristiques en entrée du système pourraient être l'objet, l'expéditeur, le message lui-même et les étiquettes « spam » ou « non-spam » ;
- **l'apprentissage non-supervisé** : les données en entrée ne sont pas catégorisées, c'est donc à l'algorithme lui-même de détecter des similarités entre ces dernières. Il s'applique généralement à des jeux de données de très grande taille et non homogènes où l'utilisation d'humains dans la préparation des données pourrait s'avérer fastidieuse ;
- **l'apprentissage mixte ou semi-supervisé** : ce type d'apprentissage tente de réaliser un compromis entre les deux types d'apprentissage présenté ci-dessus. L'algorithme est guidé au minimum avec un petit nombre d'étiquettes et se charge ensuite de classifier et de regrouper les autres caractéristiques de manière autonome.

Dans le cadre de Lectaurep, il s'agit plutôt d'apprentissage semi-supervisé qui montre de bonnes performances dans l'entraînement du modèle HTR. Le moteur HTR de *Kraken*, basé sur des réseaux de neurones, est guidé par la segmentation des zones et lignes de textes sur l'image sous la forme de coordonnées. Il se charge ensuite d'effectuer la prédiction des formes de caractères. Nous verrons par la suite les types de réseaux de neurones qui permettent de réaliser cet apprentissage.

2.1 Définir les composantes de l'intelligence artificielle dans le projet

2.1.4 La prédiction par la machine

Il s'agit de l'étape ultime qui consiste à vérifier la capacité du modèle à traiter correctement l'information. Lors de la phase d'apprentissage, le modèle tire des conclusions de ses calculs statistiques. Dans le cadre de l'évaluation du modèle HTR, il doit se rapprocher au maximum de la transcription vérité terrain réalisée sous supervision humaine. Lors de l'apprentissage, il est généralement recommandé de toujours découper son jeu de données d'apprentissage en deux (*Kraken*, configure par défaut un découpage en deux parties égales, mais il est possible de le paramétriser) :

- un set de données d'entraînement (*training set*) ;
- un set de données de test (*test set*).

Un aspect important à prendre en compte dans ces cas d'apprentissage concerne le **sur-entraînement** (*overfitting*). Le modèle a été entraîné avec des données homogènes (ensemble du *set* d'apprentissage), si bien, qu'il devient trop étroitement lié à ces dernières. Il est alors incapable de se généraliser à d'autres données.

Le cas inverse correspond au **sous-entraînement** (*underfitting*), c'est-à-dire que l'on a fourni trop peu de données pour le modèle qui est, non seulement incapable de faire des prédictions sur les données avec lesquelles il a appris (*training set*), mais qui ne peut pas non plus s'étendre à de nouvelles données.

Nous reverrons ces cas lors de la réalisation de tests de qualité de transcription présentés en partie III.

Lors de l'apprentissage les réseaux de neurones présents dans *Kraken* passent par plusieurs états (*epochs*) qui sont observables lors de son utilisation (Cf. Figure 2.4)

CHAPITRE 2 : La reconnaissance automatique des écritures dans Lectaurep : un domaine de l'intelligence artificielle et du traitement automatique du langage naturel

```

Initializing model /
stage 1/* [#####
stage 2/* [#####
stage 3/* [#####
stage 4/* [#####
stage 5/* [#####
stage 6/* [#####
stage 7/* [#####
stage 8/* [#####
stage 9/* [#####
stage 10/* [#####
stage 11/* [#####
stage 12/* [#####
stage 13/* [#####
stage 14/* [#####
stage 15/* [#####
stage 16/* [#####
stage 17/* [#####
stage 18/* [#####
stage 19/* [#####
stage 20/* [#####
stage 21/* [#####
stage 22/* [#####
stage 23/* [#####
stage 24/* [#####
stage 25/* [#####
    72/72      Accuracy report (1) 0.0000 595 595
/opt/anaconda3/envs/kraken/lib/python3.7/site-packages/coremltools/models/model.py:111: RuntimeWarning: You
ror reading protobuf spec. validator error: Input MLMultiArray to neural networks must have dimension 1 (ve
    RuntimeWarning)
stage 2/* [#####
stage 3/* [#####
stage 4/* [#####
stage 5/* [#####
stage 6/* [#####
stage 7/* [#####
stage 8/* [#####
stage 9/* [#####
stage 10/* [#####
stage 11/* [#####
stage 12/* [#####
stage 13/* [#####
stage 14/* [#####
stage 15/* [#####
stage 16/* [#####
stage 17/* [#####
stage 18/* [#####
stage 19/* [#####
stage 20/* [#####
stage 21/* [#####
stage 22/* [#####
stage 23/* [#####
stage 24/* [#####
stage 25/* [#####
    72/72      Accuracy report (2) 0.0000 595 595
    72/72      Accuracy report (3) 0.0000 595 595
    72/72      Accuracy report (4) 0.0000 595 595
    72/72      Accuracy report (5) 0.1748 595 491
    72/72      Accuracy report (6) 0.2050 595 473
    72/72      Accuracy report (7) 0.4134 595 349
    72/72      Accuracy report (8) 0.6588 595 283
    72/72      Accuracy report (9) 0.7076 595 174
    72/72      Accuracy report (10) 0.7378 595 156
    72/72      Accuracy report (11) 0.7546 595 146
    72/72      Accuracy report (12) 0.7748 595 134
    72/72      Accuracy report (13) 0.7966 595 121
    72/72      Accuracy report (14) 0.8218 595 106
    72/72      Accuracy report (15) 0.8521 595 88
    72/72      Accuracy report (16) 0.8639 595 81
    72/72      Accuracy report (17) 0.8538 595 87
    72/72      Accuracy report (18) 0.8824 595 70
    72/72      Accuracy report (19) 0.8824 595 70
    72/72      Accuracy report (20) 0.8807 595 71
    72/72      Accuracy report (21) 0.8891 595 66
    72/72      Accuracy report (22) 0.8924 595 64
    72/72      Accuracy report (23) 0.8992 595 60
    72/72      Accuracy report (24) 0.8958 595 62
    18/72 00:45:05

```

FIGURE 2.4 – Illustration des époques (*stage*) d'apprentissage pour la création d'un modèle de transcription avec Kraken ©L. Terriel, 2020, Kraken/cluster de calcul INRIA-RIOC

À chacune de ces itérations ou époques d'apprentissage, Kraken produit un modèle (format .mlmodel) et l'évalue à l'aide de métriques (*accuracy report*) qui s'appuient sur la précision (*precision*)¹³, le rappel (*recall*)¹⁴, le F1 score^{15 16}, et le coefficient de corrélation de Matthews¹⁷ (MCC). À la fin de l'entraînement, Kraken sélectionne le modèle ayant obtenu les meilleurs résultats (*best model*).

Après la phase d'apprentissage, il faut évaluer la qualité des prédictions par le modèle. Dans le cas d'un modèle HTR, on utilise des métriques comme par exemple le *Character Error Rate* (CER), ou taux d'erreur par caractères, et le *Word Error Rate* (WER), ou taux d'erreur par mots, afin de comparer la transcription obtenue automatiquement, avec la transcription de vérité terrain.

13. La **précision** (*precision*) ou valeur prédictive positive, est une mesure permettant d'estimer la proportion des items pertinents parmi l'ensemble des items proposés.

$$precision_i = \frac{\text{nb de documents correctement attribués à la classe}}{\text{nb de documents attribués à la classe}}$$

14. Le **rappel** (*recall*) ou sensibilité est la proportion des items pertinents proposés parmi l'ensemble des items pertinents.

$$rappel_i = \frac{\text{nb de documents correctement attribués à la classe}}{\text{nb de documents appartenant à la classe}}$$

15. Le F score (ou F-mesure) correspond à la moyenne harmonique de la précision et du rappel.

16. Pour plus de détails sur la précision, le rappel et le F1 score voir l'article Wikipédia, *Précision et rappel*, URL : https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel (visité le 11/09/2020)

17. Id., *Matthews correlation coefficient*, URL : https://en.wikipedia.org/w/index.php?title=Matthews_correlation_coefficient&oldid=974404329 (visité le 11/09/2020)

2.1 Définir les composantes de l'intelligence artificielle dans le projet

Nous reviendrons sur cette dernière étape d'évaluation, car il s'agit d'une de mes missions du stage. Son objectif tient en la réalisation du développement d'un outil permettant d'évaluer ces résultats à l'aide de métriques (Cf. Partie III).

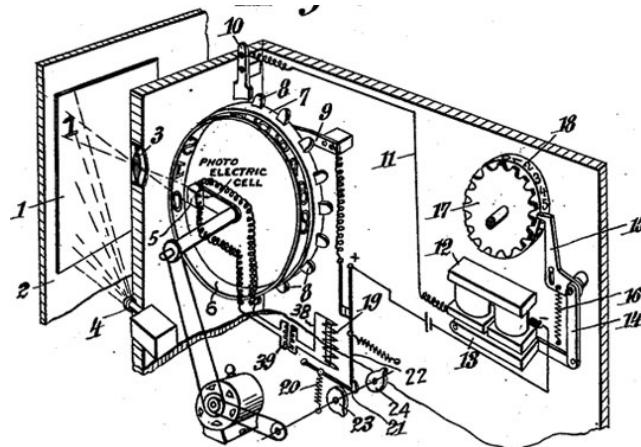
Dans la section suivante nous allons revenir sur la reconnaissance d'écritures manuscrites (HTR) et notamment sur l'usage des réseaux de neurones dans cette application de l'IA afin de clarifier certains concepts.

2.2 Avant l'intelligence artificielle, la reconnaissance optique de caractères

2.2.1 Historique

On distingue généralement la reconnaissance optique de caractères (*Optical Character Recognition - OCR*) pour les textes imprimés de la reconnaissance manuscrite de caractères (*Handwritten Transcription Recognition (HTR)*) pour les sources textuelles manuscrites.

Concernant l'HTR, on parle de reconnaissance en-ligne (*online*) quand le caractère est reconnu au moment du tracé de la forme (stylos optiques qui prennent en compte les mouvements) et de reconnaissance hors-ligne (*offline*) quand la reconnaissance du caractère s'effectue sur des caractères déjà tracés sur le papier.¹⁸.



Concevoir un système suffisamment intelligent capable de reconnaître les écritures humaines est un réel défi. Il est parfois difficile pour une personne de réussir à transcrire dès le premier regard l'écriture de quelqu'un d'autre. On peut notamment penser aux chartes anciennes, ou aux ordonnances de médecins.

La reconnaissance des écritures ne remonte pas à l'apparition des récentes applications de l'intelligence artificielle. La première invention d'un système de reconnaissance d'écriture remonte à 1929. À cette époque, Gustav Tauschek (1899-1945) crée un premier système basé sur un détecteur photosensible et un faisceau de lumière qui pointe sur un mot (Cf. Figure 2.5). La source lumineuse traverse alors des masques mécaniques (*template*) qui constituent une sorte de bibliothèque de formes de caractères stockées dans une « mémoire tambour » (également inventée par Tauschek, et ancêtre de nos disques durs actuels). Quand la lumière ne passe plus à travers le masque c'est que la forme du caractère contenu dans le mot et le caractère du masque coïncident parfaitement. Le capteur

FIGURE 2.5 – La machine à lire de Tauschek. Premier système OCR électro-mécanique. ©Patent Fletcher

18. Line Eikvil, *OCR - Optical Character Recognition*, 1993, URL : <https://www.nr.no/~eikvil/OCR.pdf>

2.2 Avant l'intelligence artificielle, la reconnaissance optique de caractères

photosensible détecte cette absence de lumière, à ce moment là, un signal est envoyé pour faire tourner le tambour d'impression à la lettre requise, et cette lettre est imprimée sur papier pour l'utilisateur. Cependant, ce principe de masque restait adapté à une quantité de fontes limitées (en cause la mémoire) et bien dessinées. Il ne permettait pas encore de caractériser l'écriture manuscrite basée sur une grande variabilité de formes.¹⁹

Durant les années 1960-1965, les premières méthodes HTR basées sur les modèles statistiques et des systèmes de classes contenant plusieurs variantes d'un même caractère émergent. L'image de la lettre et alors comparée à sa classe, et le système estime la distance statistique entre la lettre et ses variantes contenues dans la classe pour déterminer le bon caractère.

Parmi les modèles statistiques utilisés la méthode des *k plus proches voisins* (k-NN)²⁰. Le caractère binarisé est projeté dans un espace vectoriel comme une nouvelle entrée x qui est comparée à son voisinage de k (définis comme des échantillons de caractères connus). Le modèle calcule alors la distance minimum entre x et ses k voisins pour déterminer x (Cf. Figure 2.6). Cependant, cette méthode doit procéder d'un découpage univoque caractère par caractère. Ce qui rend le modèle peu utilisable pour des écritures cursives et rapprochées.

En 1974, Raymond Kurzweil, étudiant au MIT (*Massachusetts Institute of Technology*), développe un programme informatique reconnaissant des polices de caractères très différentes, adaptés aux lecteurs d'écran pour non-voyant(e)s. C'est le premier véritable succès technologique pour la reconnaissance de caractères.

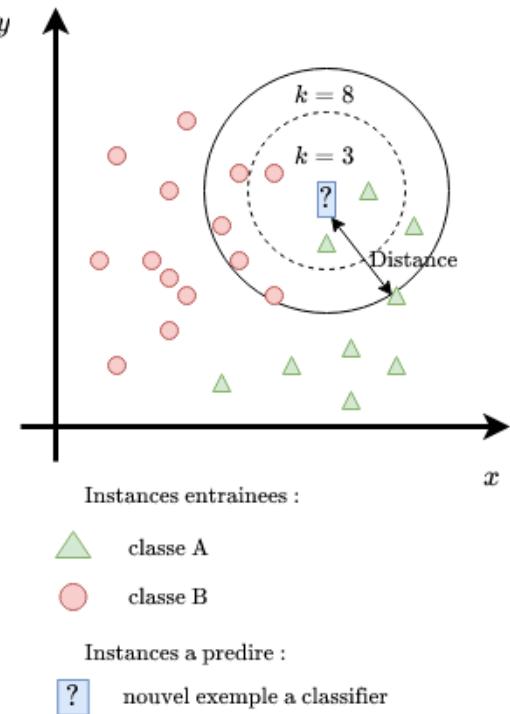


FIGURE 2.6 – Illustration simplifiée de la *méthode des k plus proches voisins* (k-NN) ©L. Terriel, 2020, Diagrams.net

19. Asma Ouji, *Segmentation et classification dans les images de documents numérisés*, thèse, INSA de Lyon, 2012, URL : <https://tel.archives-ouvertes.fr/tel-00749933> (visité le 11/09/2020)

20. Voir l'article Wikipédia, *Méthode des k plus proches voisins*, URL : https://fr.wikipedia.org/w/index.php?title=M%C3%A9thode_des_k_plus_proches_voisins&oldid=173680633 (visité le 11/09/2020)

À partir des années 1980, les modèles mathématiques comme les modèles de Markov cachés²¹ (*Hidden Markov Model*), issus des techniques de reconnaissance de la parole, sont repris pour être adaptés aux systèmes de reconnaissance des écritures. Les résultats obtenus sont des succès. Combinés à des modèles linguistiques pour reconnaître la position des parties dans la phrase, et des méthodes de segmentation avancées, afin d'isoler dans l'image les lignes de texte et les caractères à l'intérieur de celles-ci. Ces modèles linguistiques utilisés en *post-traitement* se basent sur des dictionnaires de mots, de syllabes, de *N-grammes*²² qui réalisent des estimations du nombre d'occurrences de sous-séquences de mots.

Les années 1990, sont marquées par le retour des technologies de réseaux de neurones. Les systèmes de reconnaissance de caractères s'améliorent nettement en combinaison avec des classificateurs statistiques. Ces méthodes hybrides permettent alors à des entreprises comme IBM, Toshiba ou Hitachi de créer des chaînes de traitement à grande échelle. Nous pouvons prendre comme exemple le cas du tri postal avec la lecture automatique des codes postaux ou encore les banques avec la détection automatique des zones d'un chèque.

21. Voir l'article Id., *Modèle de Markov caché*, URL : https://fr.wikipedia.org/w/index.php?title=Mod%C3%A8le_de_Markov_cach%C3%A9&oldid=163612626 (visité le 11/09/2020)

22. un *N-gramme* est une séquence de taille n , piochée dans une séquence de taille plus grande que n . Ainsi dans « Bonjour », « Bo » est un bi-gramme de « Bonjour », « Bon » est un tri-gramme de « Bonjour » etc. Voir l'article Id., *N-gramme*, URL : <https://fr.wikipedia.org/w/index.php?title=N-gramme&oldid=167585561> (visité le 23/09/2020)

2.2.2 Modèles de réseaux de neurones profonds appliqués à l'HTR

2.2.2.1 Historique des réseaux de neurones

La reconnaissance de formes est l'un des enjeux de l'IA. Actuellement, les systèmes HTR les plus performants reposent sur la technologie des réseaux de neurones (*Google Tesseract-OCR* ou *Kraken* par exemple). La démocratisation de leur utilisation à travers des bibliothèques de code comme *Tensorflow* et *Keras* en langage Python, voient leurs implémentations facilitées. Il s'agit de puissants algorithmes basés sur l'analogie avec la neurobiologie qui peuvent réaliser des tâches et des prédictions dans le contexte pour lequel ils ont été entraînés et cela de manière autonome.

En 1943, Warren McCulloch (1898-1969) un neurologue et Walter Pitts (1923 - 1969) un logicien, proposent dans l'article « *A Logical Calculus of Ideas Immanent in Nervous Activity* »²³ une première représentation du neurone formel. Le neurone y est décrit comme un « automate à seuil, dont l'état actif ou non, désigne une valeur logique, vraie ou fausse »²⁴.

Frank Rosenblatt (1928-1971) applique, en 1958, le concept de neurone formel de McCulloch-Pitts en réseau pour simuler le fonctionnement rétinien reconnaissant des formes. La machine de Rosenblatt ou *perceptron* est la première implémentation de réseaux de neurones muni d'une règle d'apprentissage simple²⁵.

Ces approches dites « connexionistes » vont perdurer jusqu'à atteindre, vers 1970, leurs limites en terme de connaissance scientifique et de puissance de calcul pour l'époque. C'est encore le début de l'architecture « Von Neumann »²⁶ pour les ordinateurs qui restent limités en nombre et qui demandent beaucoup de main-d'œuvre.

Dans les années 1980, des avancées théoriques majeures améliorent considérablement l'approche des réseaux de neurones comme l'estimation du gradient par rétro-propagation de l'erreur (1989) et l'analogie des phases d'apprentissages avec les modèles markoviens cachés (1982), qui relancent le courant « connexioniste ».

23. Warren S. McCulloch et Walter Pitts, « A logical calculus of the ideas immanent in nervous activity », *The bulletin of mathematical biophysics*, 5–4 (1943), p. 115-133, URL : <https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf> (visité le 11/09/2020)

24. Francisco Varela, *Invitation aux sciences cognitives*, Seuil, 1996, pp.

25. F. Rosenblatt, « The perceptron : a probabilistic model for information storage and organization in the brain. » *Psychological review*, 65–6 (1958), p. 386-408, URL : <https://psycnet.apa.org/record/1959-09865-001>

26. modèle conceptuel du fonctionnement d'un ordinateur actuel avec la division : une unité de contrôle, une unité arithmétique et logique, une mémoire et des entrées et des sorties.

Les années 1990 sont alors marquées par une amélioration considérable de la puissance des machines grâce aux technologies GPU (*Graphics Processing Unit* ou processeurs graphiques) qui permettent d'accélérer les calculs, ainsi qu'aux nouveaux algorithmes qui réalisent l'estimation de millions de paramètres du *perceptron* et créés de nombreuses couches de neurones formels (*layers*) aux fonctions spécifiques. L'accroissement exponentiel des bases de données permettent de tester ces réseaux de neurones avec des données toujours plus nombreuses.

2.2.2.2 Du neurone formel aux réseaux de neurones multi-couches : fonctionnement

Un neurone formel est à la fois une représentation mathématique et un algorithme informatique basé sur le neurone biologique. On considère généralement un neurone formel qui possède plusieurs entrées x_1, x_2, \dots (dendrites) et une sortie y (point de départ de l'axone). L'excitation du synapse est généralement représenté par des coefficients numériques ou poids synaptiques notés w . Au cours de l'apprentissage ces coefficients sont ajustés automatiquement. Par exemple, l'algorithme de rétropropagation du gradient, cherche à minimiser le gradient d'erreur, qui est une correction et un ajustement des poids synaptiques. Le neurone simple, comme le *perceptron*, réalise la somme pondérée des entrées : données et poids synaptiques. Le résultat obtenu est passé dans une fonction d'activation ou de transition ($f(n)$), souvent non linéaire. Les fonctions d'activation sont, par exemple, les fonctions sigmoïdes, ReLU, *Softmax* etc. qui s'appuient sur des modèles mathématiques.

On définit généralement un seuil d'activation qui est également un coefficient numérique paramétrable noté le plus souvent w_0 ou θ . Si la valeur finale obtenue dépasse ce seuil d'activation, le neurone génère une nouvelle sortie y . La Figure 2.7 résume ce fonctionnement.

2.2 Avant l'intelligence artificielle, la reconnaissance optique de caractères

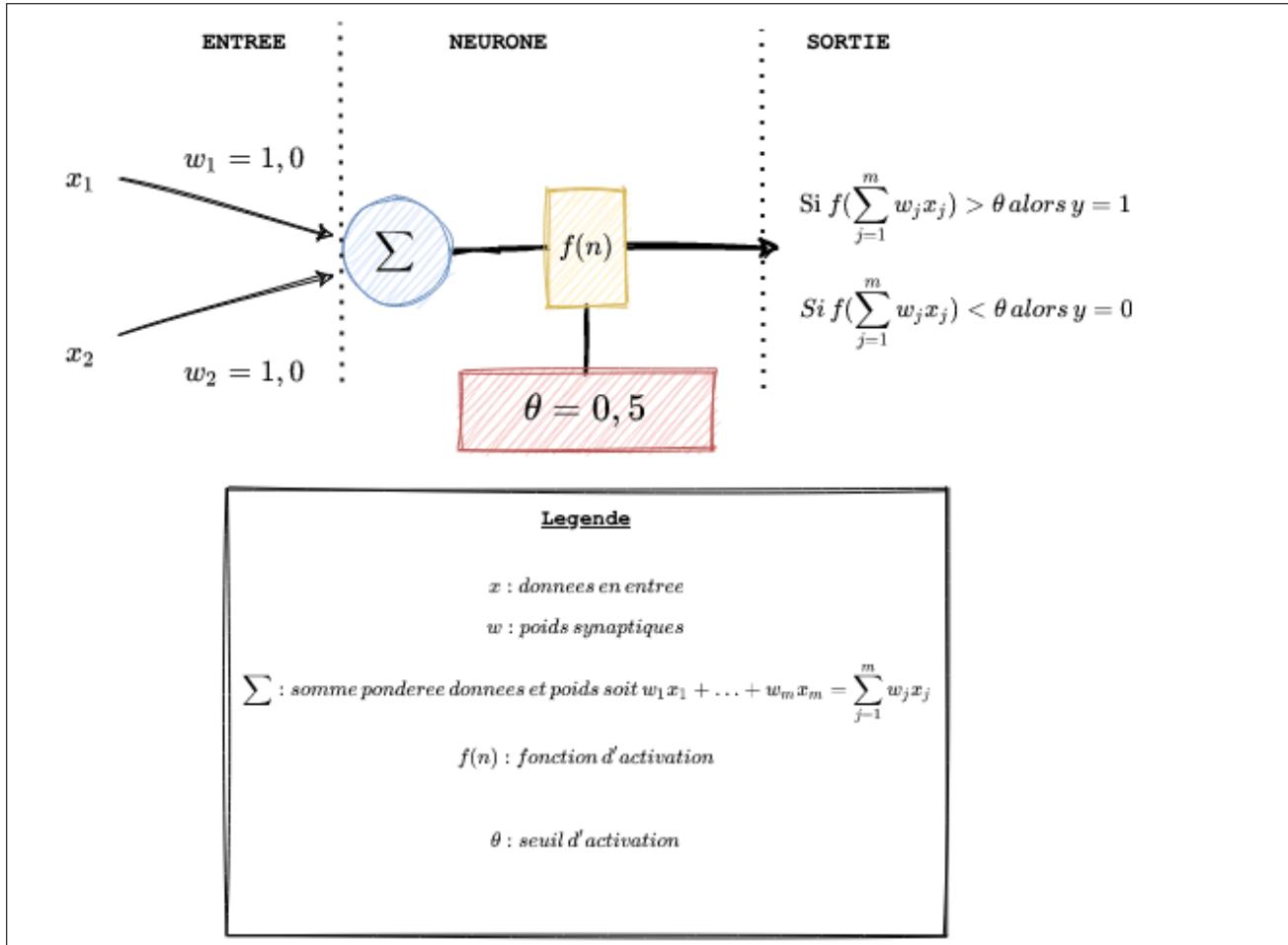


FIGURE 2.7 – Illustration simplifiée d'un neurone formel ©L. Terriel, 2020, Diagrams.net

Le neurone formel est une unité élémentaire dans un réseau de neurones complet. Un réseau de neurones peut donc être représenté sous la forme d'un graphe qui relie plusieurs neurones formels qui comprennent parfois des propriétés ou des fonctions d'activation différentes, et qui forment une ou plusieurs couches cachées (*Hidden layers*), on parle aussi de réseaux de neurones multi-couches. Ces couches sont généralement définies selon le contexte d'application de ces réseaux comme la reconnaissance de forme, cela procède souvent d'un long travail d'expérimentation même si actuellement des standards se dégagent (Cf. Figure 2.9).

Dans ces réseaux, l'information circule en avant grâce à des algorithmes de propagation en avant (*feed forward propagation*) et des algorithmes de propagation en arrière qui ont des fonctions correctrices (*backward propagation*) des poids synaptiques sur les couches les plus responsables des erreurs. Tout le travail d'ajustement des réseaux pour générer la sortie voulue se situe dans la capacité à corriger les poids synaptiques²⁷.

27. Pour une autre approche des réseaux de neurones on pourra visionner la présentation Mathieu Aubry (École des Ponts ParisTech), « IA et apprentissage automatique : des outils pour l'analyse et la valorisation du patrimoine », Journée d'étude « Intelligence artificielle et institutions patrimoniales » de

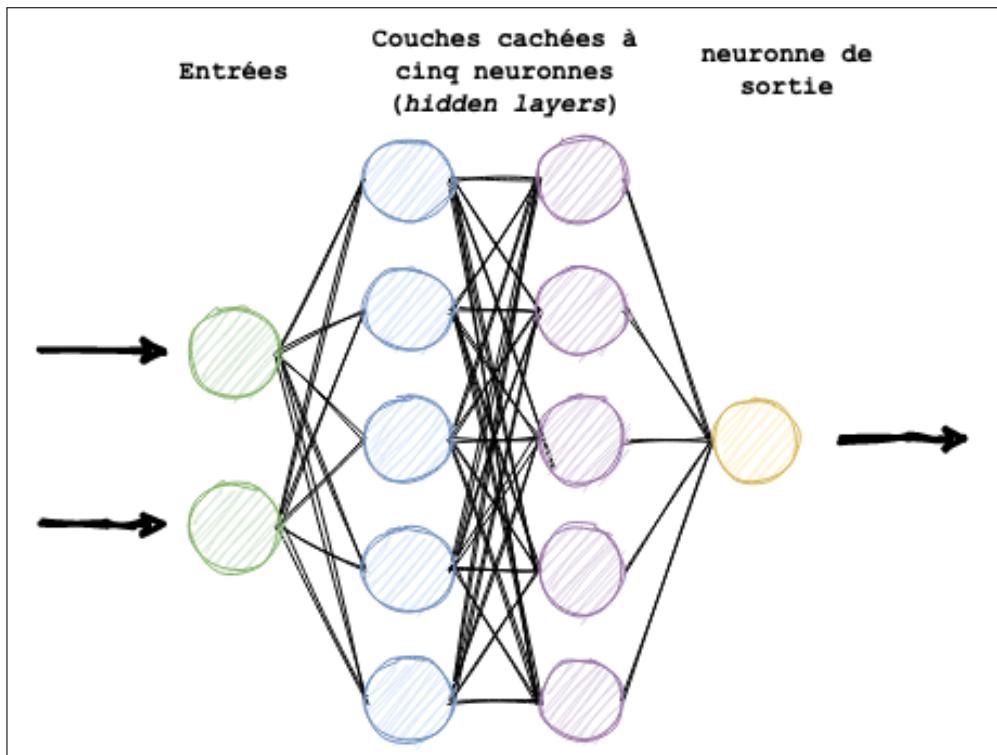


FIGURE 2.8 – Illustration simplifiée d'un réseau de neurones à deux couches de neurones cachées ©L. Terriel, 2020, Diagrams.net

2.2.2.3 Quel réseau de neurones pour l'HTR ?

Il existe plusieurs sortes d'architectures pour programmer des réseaux de neurones, variant en fonction du contextes d'utilisation. Dans le cadre de la reconnaissance de formes sont utilisés principalement des architectures de réseaux de neurones dites « mixtes ». Ces dernières utilisent des réseaux de neurones récurrents (RNN ou *Recurrent Neural Network*) chargés de modéliser les séquences de caractères et des réseaux de neurones à convolutions qui quant à eux sont chargés d'extraire des caractéristiques (traits, caractères, mots, amas de pixels dans l'image etc.).

Ces réseaux hybrides sont particulièrement adaptés aux traitements des séquences temporelles comme les phrases qui sont des suites de formes s'enchaînant pour constituer des mots et demandent de se souvenir des enchaînements de formes précédentes. Parmi les RNN, un type particulier de réseaux de neurones appelé LSTM (*Long short-term memory*) permet de doter le neurone formel de boucles qui lui offrent une capacité de mémorisation des enchaînements de formes. Ce réseau hybride est implémenté dans *Kraken* (Figure 2.9) par le biais de la bibliothèque en langage C++ *CLSTM* qui fonctionne en Python comme une extension²⁸.

l'ADEMEC, Bibliothèque nationale de France, 11 décembre 2019, URL : https://www.youtube.com/watch?v=MvaXQ2t2mPs&list=PLayqwLSo_nPW1wHnVw-gJPwMya4gYBPe0&index=2

28. CLSTM, est une implémentation des RNN de type LSTM, voir : <https://github.com/tmbdev/clstm>

2.2 Avant l'intelligence artificielle, la reconnaissance optique de caractères

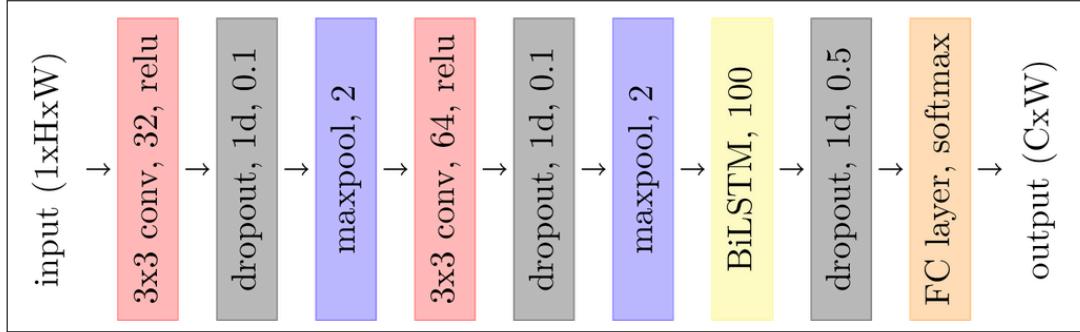


FIGURE 2.9 – Architecture hybride des différentes couches disposant de RNN (BiLSTM) et de neurones à convolution, utilisée dans *Kraken* ©KIESSLING (Benjamin), *Kraken - an Universal Text Recognizer for the Humanities*, DH2019, 2019, URL : <https://dev.clariah.nl/files/dh2019/boa/0673.html> (visité le 11/09/2020)

Ces réseaux de neurones sont très gourmands en ressources matérielles. En effet, ils demandent d'effectuer de très nombreux calculs matriciels lors de l'entraînement des modèles, à l'instar des jeux vidéos actuels en haute définition. En l'absence de machines dédiées durant le stage, nous avons bénéficié d'un accès au *cluster* de calcul d'INRIA (RIOC) pour l'entraînement des modèles de transcription et de segmentation sur *Kraken*. Un cluster de calcul s'apparente à un ordinateur distant (accès sans interface graphique) qui dispose de plusieurs GPU, permettant d'accélérer des traitements informatiques lourds.

Pour donner un exemple, un ordinateur classique avec un CPU (*central processing unit* ou processeur central) réalise un entraînement en trois à quatre heures, le *cluster* de calcul réduit à quelques minutes cette charge de travail. L'accès à ce *cluster* s'effectue via le terminal de l'ordinateur à l'aide de lignes de commandes en BASH²⁹, permettant une connexion à distance sécurisée (protocole SSH (*Secure Shell*)) en sus d'un VPN (*Virtual Private Network* ou réseau privé virtuel) (Cf. Figure 2.4).

29. **BASH** (*Bourne-Again shell*) est une interface en ligne de commande *open-source* de type *script* (shell UNIX du projet GNU). Fournit une liste de commandes pour interagir avec l'ordinateur (copie de fichiers, lancement de programme etc.), possibilité de regrouper les commandes dans un fichier (*script*), vérification des commandes interprétées, et interprétation des commandes.

2.3 Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel

Dans un projet de reconnaissance automatique d'écritures manuscrites, l'extraction du texte n'est pas tout, il s'agit là d'une brique qui doit produire une bonne transcription pour permettre des exploitations spécifiques sur ce texte.

La reconnaissance automatique peut être conçue comme une *pipeline* ou un pont entre l'obtention de données textuelles de qualité et des outils qui permettent leurs compréhension. C'est outils sont développés dans le domaine du traitement automatique du langage naturel (TAL ou TALN, en anglais *Natural Language Processing* (NLP)).

Le TAL est une sous-branche de l'IA, pouvant s'appuyer sur les techniques de ML et de DL expliquées plus haut. Ce domaine traite les langues parlées par les humains. Le TAL couvre donc un large panorama d'applications comme la génération automatique de résumé de texte, la traduction, la correction, la recherche d'information ou encore la « compréhension » d'un texte : analyse sémantique et syntaxique, visant déterminer le rôle de chaque mot dans une phrase, par exemple. Les tâches techniques du TAL passent par la reconnaissance d'entités nommées, l'extraction de relations ou encore l'analyse des mots.

Dans cette section, nous aborderons, succinctement, les techniques classiques propres aux TAL. Dans un second volet nous nous attarderons sur les applications de ces techniques dans le projet Lectaurep à destination des publics des archives, des chercheuses, des chercheurs et des archivistes.

2.3 Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel

2.3.1 Les techniques du TAL

La tokenisation - C'est le traitement qui permet de découper un texte en phrases, une phrase en mots ou un mot en caractères. On parle alors de *tokens* pour ces unités récupérées (Cf. Figure 2.10).

```
basic : ['Déjà', 'M', '.', 'Lidenbrock', '!', 's', 'écria', 'la', 'bonne',
'dîner', 'a', 'le', 'droit', 'de', 'ne', 'point', 'être', 'cuit', ',', 'c
i', 'M', '.', 'Lidenbrock', 'rentre', '-', 't', 'il', '?', 'Il', 'nou
', '.', 'Et', 'la', 'bonne', 'Marthe', 'regagna', 'son', 'laboratoire', 'c
tokens_ref_nltk : ['-','Déjà', 'M', 'Lidenbrock', 's', '!', 'écria', 'la'
'a', 'le', 'droit', 'de', 'ne', 'point', 'être', 'cuit', 'car', 'il', 'n',
', 'rentre', 't', 'il', 'il', 'Il', 'nous', 'le', 'dira', 'vraisemblablemen
'son', 'laboratoire', 'culinaire']
```

FIGURE 2.10 – Exemple d'une tokenisation en mots réalisée avec un tokenizer développé à partir des expressions régulières (Regex) et le tokenizer du package Python NLTK ©L. TERRIEL, 2020, Pycharm

Supprimer les mots les plus fréquents - Il s'agit généralement de retirer les mots vides (*stop words*) dans un texte. On considère comme mots vides les occurrences communes qu'il est inutile d'indexer ou d'utiliser car elles peuvent brouter une recherche ou fausser des résultats ; cette suppression est paramétrable. Par exemple, « le », « la », « de », ou « du » sont des exemples de *stop words*.

La racinisation³⁰ - Elle consiste à réduire un mot à sa « racine ». Le but du la racinisation est de regrouper de nombreuses variantes d'un mot comme une seule et même unité. Par exemple, la technique s'appliquant sur « contrat » et « contrats » permet de faire ressortir un mot équivalent.

La reconnaissance d'entités nommées - En anglais *Named-entity recognition* ou NER, elle cherche à extraire les entités telles que des noms de personnes, des noms de lieux ou tout autre information pertinente pour la compréhension approfondie du texte. Certaines applications permettent de visualiser ces entités nommées sous la forme de mots étiquetés sur l'écran (*visualizers*). L'outil Entity-fishing³¹, disponible en ligne, offre la possibilité d'extraire les entités nommées d'un texte, de les relier à un référentiel (ici Wikipédia) et de visualiser l'étiquetage des mots. L'étiquetage de ces informations peut être réalisé dans un format structuré en JSON³² (comme dans *Entity-fishing*) ou en XML (Cf. Figure 2.11).

30. ou désuffixation (*stemming*)

31. Patrice Lopez, *Entity-fishing - Entity Recognition and Disambiguation*, URL : <http://nerd.huma-num.fr/nerd/>

32. **JSON** (*JavaScript Object Notation*) est un format de structuration des données dérivé du langage informatique Javascript.

CHAPITRE 2 : La reconnaissance automatique des écritures dans Lectaurep : un domaine de l'intelligence artificielle et du traitement automatique du langage naturel

An 1920, mois de **MAI BOUYGE**, à sa fe à **PARIS** quai des Gds **AUGUSTINS** 45 - d'inscription au **TAL** de Ce de la **SEINE** du 6 **XBRE** 1918 N° 84047 contre **VEUVE RENAI**s, dt à **PARIS** rue St andré des arts 34 1510 34 1010 14 Inventaire Lautard (à **PARIS** **RUE GALANDE** 59 ou demeurait et ou est dececé le 2 mars 1920 **JEAN JOSEPH**) époux de **MARIE LÉONIE JOSEPHINE MAJOREL** 15 7.50 1011 15 procuration Coache (par **LÉONARD ADOLPHE MARIUS**) et **JOSEPHINE HENRIETTE SCHMIDIGER**, sa fe à **PARIS** **RUE DIDOT** 121, et autre, en blanc, pour vendre

PARIS

Type: **LOCATION**
Normalized: Paris
Domains: **Administration, Sociology**
conf: 0.8139



Paris (prononcé) est la capitale de la France. Elle se situe au cœur d'un vaste bassin sédimentaire aux sols fertiles et au climat tempéré, le **bassin parisien**, sur une boucle de la **Seine**, entre les confluents de celle-ci avec la **Marne** et l'**Oise**. Ses habitants s'appellent les Parisiens. Paris est également le **chef-lieu** de la **région Île-de-France** et l'unique **commune française** qui est en même temps un **département**. Commune centrale de la **Métropole du Grand Paris**, créée en 2016, elle est divisée en **arrondissement**, comme les villes de **Lyon** et de **Marseille**, au nombre de vingt. L'État y dispose de prérogatives particulières exercées par le **préfet de police de Paris**.

Wikidata statements

▼

References:  

▼

```
{
  "runtime": 614,
  "nbest": false,
  "text": "An 1920, mois de Mai Bouyge, à sa fe à Paris quai des Gds augustins 45 - d'inscription au Tal de Ce de la seine du 6 xbre 1918 N° 84047 contre Veuve Renais, dt à Paris rue St andré des arts 34 1510 34 1010 14 Inventaire Lautard (à Paris rue Galande 59 ou demeurait et ou est dececé le 2 mars 1920 Jean Joseph) époux de Marie Léonie Josephine Majorel 15 7.50 1011 15 procuration Coache (par Léonard adolphe Marius) et Josephine Henriette schmidiger, sa fe à Paris rue Didot 121, et autre, en blanc, pour vendre",
  "language": {
    "lang": "fr",
    "conf": 0
  },
  "global_categories": [
    {
      "weight": 0.04819277108433738,
      "source": "wikipedia-fr",
      "category": "Changement de nom de ville dans l'Histoire",
      "page_id": 9750163
    },
    {
      "weight": 0.012048192771084345,
      "source": "wikipedia-fr",
      "category": "Doggerland",
      "page_id": 7327787
    }
  ]
}
```

FIGURE 2.11 – Extrait d'un répertoire du notaire Marotte passé dans l'outil *Entity-fishing* pour l'étiquetage des entités nommées et sa réponse en JSON comprenant le taux de confiance (*weight*) accordé pour chaque candidats du référentiel Wikipédia. ©L. TER-RIEL, 2020, *Entity-fishing*.

2.3 Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel

L'étiquetage morpho-syntaxique - Part-of-Speech Tagging (POS) en anglais, consiste à étiqueter la fonction grammaticale de chaque mot dans une phrase. On peut également visualiser ces étiquettes et leurs dépendances (Cf. Figure 2.12).

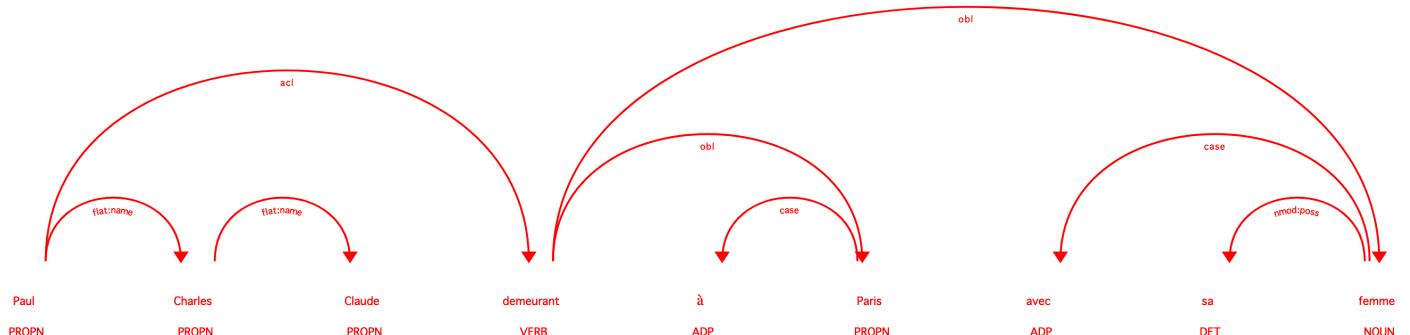


FIGURE 2.12 – Exemple de *Part-of-Speech Tagging* réalisé sur un document de vérité terrain du répertoire de notaire Marotte. Réalisé avec un *script Python* (Cf. Annexes, Figure F.1) ©L. TERRIEL, 2020, *Pycharm*.

Plongement de mots ou lexical

- Appelé aussi *Word embedding* en anglais, il s'agit d'une méthode qui consiste à représenter les mots sous forme de vecteurs (Cf. Figure 2.13). Cette technique permet notamment de récupérer des valeurs numériques. Ainsi des mots apparaissant dans un même contexte auront des chances d'avoir des vecteurs similaires. On l'utilise notamment pour évaluer la similité entre des mots ou des phrases grâce aux distances statistiques entre ces vecteurs (par exemple, la distance euclidienne) ou des métriques comme la mesure cosinus (Cf. Partie III).

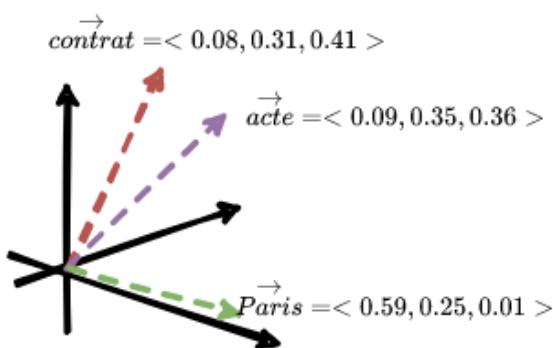


FIGURE 2.13 – Exemple simplifié de représentation vectorielle des mots (*Word embedding*). Les mots « contrat » et « acte » ont des vecteurs proches (vecteurs colinéaires) tandis que le mot « Paris » présente un vecteur éloigné par rapport à ses pairs (vecteurs orthogonaux). ©L. Terriel, 2020, *Diagrams.net*

Modèle Transformer - Plus récents, les modèles *Transformer* sont des modèles de langues entraînés à partir de RNN, comme BERT³³, qui sont mis à disposition pour des applications telles que la génération de texte, ou encore la bonne prédiction d'une suite de phrases, et des tâches de question/réponses. BERT dispose de variantes dans d'autres langues, comme CamemBERT. C'est un modèle pour la langue française qui a été entraîné par l'équipe d'ALMANaCH avec 130 *gigabits* de données textuels³⁴.

33. BERT (*Bidirectional Encoder Representations from Transformers*) est un modèle de langage créé par Google, il existe en version multi-langue et a été entraîné sur Wikipedia de plus de 104 langues.

34. Éric Villemonte de la Clergerie, Yoann Dupont, Louis Martin, Benjamin Muller, Laurent Romary,

Dans la partie III, nous verrons que certaines de ces techniques de TAL nous ont été utiles pour l'élaboration de métriques dans le cadre du développement de l'application d'évaluation des transcriptions *Kraken-Benchmark*.

2.3.2 Applications et potentialités du TAL pour Lectaurep

Les tâches du TAL décrites plus haut peuvent être mises au service du projet Lectaurep et plus précisément des résultats des transcriptions obtenues par le modèle HTR.

Correction des résultats HTR - Dans un premier cas, ces techniques peuvent permettre d'améliorer les résultats HTR. Ceci peut être réalisé par une combinaison de détection des erreurs de transcriptions et de correcteurs orthographiques en sortie de l'HTR pour corriger les textes de répertoires bruités³⁵.

Extraction d'entités nommées et structuration en XML TEI - L'extraction d'entités nommées dans les répertoires de notaires peut permettre de nombreuses exploitations pour les chercheurs et les publics des archives. Les points suivants développent cet aspect. Cependant, comme nous l'avons vu plus haut, cela présuppose de disposer d'un format pour structurer les données, comme le JSON ou le XML (comme dans le cas d'*Entity-fishing*). Les *Guidelines P5*³⁶ de la TEI proposent un ensemble de balises qui peuvent permettre d'étiqueter sémantiquement les entités nommées repérées dans les répertoires de notaires³⁷ (noms de personnes, dates, biens, types d'actes etc.).

Faciliter la recherche dans les répertoires - L'extraction d'entités nommées est un moyen d'optimiser l'accès aux corpus d'actes notariés pour les chercheuses, chercheurs et le public des services d'archives qui formulent des requêtes dans les moteurs de recherche dédiés. On distingue généralement plusieurs types de recherches :

Benoît Sagot, Djamel Seddah, Pedro Javier Ortiz Suárez, **BERT**, a Tasty French Language Model, URL : <https://arxiv.org/abs/1911.03894>

35. Marion Baranes, « Vers la correction automatique de textes bruités : Architecture générale et détermination de la langue d'un mot inconnu », dans *RECITAL'2012 - Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, 2012, URL : <https://hal.inria.fr/hal-00701400> (visité le 15/09/2020) ; Thibault Magallon, Frédéric Béchet et Benoit Favre, « Détection d'erreurs dans des transcriptions OCR de documents historiques par réseaux de neurones récurrents multi-niveau », dans *25e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, Rennes, France, 2018, URL : <https://hal.archives-ouvertes.fr/hal-01905258> (visité le 15/09/2020)

36. TEI Consortium, *P5 : Guidelines for Electronic Text Encoding and Interchange*, 2020, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/index.html> (visité le 10/09/2020)

37. Pablo Ruiz, *Concept-based and relation-based corpus navigation : applications of natural language processing in digital humanities*, thèse, PSL Research University, 2017, URL : <https://tel.archives-ouvertes.fr/tel-01575167> (visité le 15/09/2020)

2.3 Extraire, analyser et exploiter les données de l’HTR avec le traitement automatique du langage naturel

- La recherche en texte intégral (en texte libre ou plein-texte) : c'est la technique classique d'accès à un mot dans un document électronique. Le moteur de recherche examine tous les mots enregistrés dans le document, et essaye de les faire correspondre à la requête de l'utilisateur ;
- La recherche approximative (ou floue ou *fuzzy search*) : il s'agit de trouver un motif (*pattern*) approximatif plutôt qu'une correspondance exacte avec la requête de l'utilisateur composée de sous-chaînes de caractères. Il s'agit donc de proposer des suggestions de recherche ou des corrections orthographiques, plutôt que de répondre directement à la requête de l'utilisateur, qui propose souvent une idée générale plutôt qu'une étiquette technique précise.

Par exemple, si l'utilisateur effectue une recherche avec le mot-clé « mariage » le moteur pourra suggérer une formulation plus précise comme « contrat de mariage ». Cependant, cela demande en amont de calibrer correctement la sensibilité du système afin d'évaluer le niveau d'approximation maximal que l'utilisateur peut proposer lors de sa requête. Les algorithmes utilisés pour ce type de recherche s'appuient généralement sur les distances d'édition (comme la distance de Levenshtein) et les métriques de similarité, sur lesquelles nous reviendrons dans la partie III ;

- La recherche par mots-clés : le principe consiste à extraire certains mots récurrents ou stratégiques (entités nommées) dans les documents. Cette recherche s'appuient sur les problèmes de repérage des mots-clés ou, en anglais, de *Keyword spotting*³⁸. Cela permet d'identifier rapidement dans les requêtes particulièrement longues d'un utilisateur les mots-clés et de proposer rapidement les résultats. Cette utilisation peut également constituer une stratégie dans l'élaboration d'une recherche à facettes.

38. Sur les aspects de *Keyword spotting* consulter M.L. Bonhomme, *Défis et opportunités de la reconnaissance automatique d’écriture manuscrite pour les documents d’archives : l’exemple des réertoires des notaires de Paris...*, pp.53-55. Notons que le *Keyword spotting* peut également s'appliquer à la recherche de motifs récurrents dans les images (*query by example*). Isabella di Lenardo (EPFL/INHA/projet Venice Time Machine), « Chercher dans les grands corpus d’images à travers l’Intelligence Artificielle : défis et résultats », Journée d’étude « Intelligence artificielle et institutions patrimoniales » de l’ADEMEC, Bibliothèque nationale de France, 11 décembre 2019, URL : https://www.youtube.com/watch?v=ndT3NLPeQFM&list=PLayqwLSo_nPW1wHnVw-gJPwMya4gYBPe0&index=3

Liage des entités nommées à d'autres référentiels de données - Comme nous l'avons déjà évoqué, un extracteur d'entités nommées peut être configuré pour étiqueter mais aussi relier les entités issues des répertoires à des référentiels³⁹ internes aux Archives nationales, ou externes sous la forme de bases de données présentes sur le *web*. Ainsi un outil comme *Entity-fishing* est paramétrable par le biais de son API ou de son code source pour relier des entités à des référentiels de données du *web* (par défaut, Wikipédia).

Ces entités nommées peuvent être reliées à partir de liens à des bases de données contenant des informations de nature géographiques (GeoNames⁴⁰), documentaires (data.bnf.fr⁴¹ ou Projet Gutenberg⁴²), encyclopédiques (Wikidata⁴³), vocabulaire multilingue (Eurovoc⁴⁴) etc. C'est une manière de faire entrer le projet dans le cadre des « données ouvertes liées » (*linked open data*), en rattachant les informations contenues dans les répertoires au *web* sémantique. Cela dans l'optique d'augmenter la pertinence des recherches des utilisateurs de Lectaurep, d'ouvrir la base de connaissances autour des répertoires de notaires et de permettre d'autres axes de lecture de ces sources historiques.

Les Archives nationales explorent actuellement la possibilité de relier leurs données aux ressources du *web*. À travers le nouveau modèle conceptuel RiC-CM (*Records in Contexts - Conceptual Model*) décrivant le monde des archives comme un graphe de données et RiC-O⁴⁵ (*Records in Contexts - Ontology*) qui est la représentation technique du modèle conceptuel, il s'agit de relier les informations contenues dans les notices producteurs (NP) en format EAC-CPF et les instruments de recherche (IR) en EAD entre-eux et à des dépôts extérieurs de métadonnées (référentiels) sous la forme de liens RDF⁴⁷. Une première preuve de concept, PIAAF⁴⁸ (Pilote d'interopérabilité pour les Autorités Archivistiques françaises) à permis d'expérimenter la visualisation des fonds des AN sous la forme d'un graphe orienté pour permettre une meilleure interopérabilité des données de description des fonds avec le *web*. Le projet se dote petit à petit d'un outil de conversion des métadonnées contenues dans les IR et NP en RDF et favorise l'enrichissement des

39. Un **référentiel** est un moyen de rassembler des connaissances d'un certain type dans des formats structurés (JSON, SQL, ou XML) et normés (TEI, EAD, EAC-CPF, MARC, *Dublin Core* etc.) (taxinomie, ontologie, thésaurus ou vocabulaire contrôlé).

40. Geonames, URL : <https://www.geonames.org/>

41. data.bnf.fr, URL : <https://data.bnf.fr/>

42. Projet Gutenberg, URL : <https://www.gutenberg.org/browse/languages/fr>

43. Wikidata, URL : https://www.wikidata.org/wiki/Wikidata:Main_Page

44. Eurovoc, URL : <https://eur-lex.europa.eu/browse/eurovoc.html?locale=fr>

45. RiC (*Records in Context*) est un standard de description récent (2020, version 1.0) des archives fondé sur l'intuition des nouvelles pratiques de recherche des utilisateurs en contexte numérique. C'est une ontologie⁴⁶ qui doit permettre de relier les ressources du *web* avec les référentiels des Archives nationales. URL : https://www.ica.org/standards/RiC/RiC-0_v0-1.html

47. **Ressource Description Framework** (RDF) est un modèle de graphe et une grammaire du W3C destiné à décrire formellement les ressources *Web* et leurs métadonnées, afin de permettre le traitement automatique de telles descriptions et de les faire correspondre entre-elles.

48. PIAAF, URL : <https://piaaf.demo.logilab.fr/>

2.3 Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel

référentiels des AN⁴⁹. À noter qu'en mars 2016, au forum des archivistes, une démonstration de graphe contenant des informations IR et NP du DMC avait été réalisée⁵⁰.

Une nouvelle approche pour l'histoire notariale - L'historiographie a soulevé, très tôt, l'intérêt de l'usage des archives notariales pour l'histoire des personnes (sources prosopographiques), l'histoire d'un quartier ou l'histoire des familles. Puis dans des cadres plus étendus comme l'histoire économique et financière, l'histoire des mentalités ou encore l'histoire administrative, sous l'angle des données quantitatives. L'École des Annales soulignait l'intérêt de ces sources pour l'étude des dynamiques socio-économiques de l'histoire de France sur le temps long :

Avec celles de l'Enregistrement, les archives notariales figurent les sources monographiques qui se prêtent le mieux à ces doubles études, particulières et générales, statiques et dynamiques. Elles couvrent une longue période de l'histoire de France, s'étendent à tout son territoire, et, malgré la variété des règles et coutumes juridiques, présentent une unité certaine qui facilite et autorise les rapprochements.⁵¹

De nombreux historiens ont fait de ces documents la matière première de leurs réflexions : Jean-Paul Poisson (1920-2005) prenant appui sur les actes notariés pour évaluer leurs potentialités pour étudier les tendances des populations⁵², Philippe Ariès (1914-1984) qui faisait usage des testaments dans ses études sur la perception de la mort⁵³ ou encore Roland Mousnier (1907-1993) qui a procédé à des travaux de dépouillement et d'analyses des contrats de mariage du XVII^e et du XVIII^e siècle. Dans une instruction des Archives de France du 16 décembre 2009, l'abaissement des délais de communicabilité des archives de cent à soixantequinze ans prend appui sur l'intérêt de ces sources pour l'histoire quantitative :

49. Florence Clavaud et Pauline Charbonnier, *Records in Contexts aux Archives nationales : enjeux et premières réalisations*, hypotheses.org, 2020, URL : https://f.hypotheses.org/wp-content/blogs.dir/2167/files/2020/02/20200128_3_RiCauxAN_EnjeuxPremieresRealisations.pdf et Anila Angjeli, F. Clavaud et Stéphanie Roussel, « Représenter en RDF, interconnecter et visualiser en graphe des jeux de métadonnées archivistiques de provenances multiples : un projet de prototype », *Gazette des archives*, 245-1 (2017), p. 157-171, URL : https://www.persee.fr/doc/gazar_0016-5522_2017_num_245_1_5523 (visité le 21/09/2020)

50. F. Clavaud et Cyprien Henry, *Vers un référentiel national des notaires ?*, forum des archivistes, 2016, URL : <https://fdocuments.fr/document/relier-donnees-referentielnotaireschenryfclavaud-final.html> (visité le 21/09/2020)

51. Adeline Daumard et François Furet, « Méthodes de l'Histoire sociale : les Archives notariales et la Mécanographie », *Annales*, 14-4 (1959), p. 676-693, URL : https://www.persee.fr/doc/ahess_0395-2649_1959_num_14_4_2865 (visité le 21/09/2020), pp.676-677.

52. Jean-Paul Poisson, « Histoire des populations et actes notariés », *Annales de Démographie Historique*, 1974-1 (1974), p. 51-57, URL : https://www.persee.fr/doc/adh_0066-2062_1974_num_1974_1_1229 (visité le 21/09/2020)

53. Alain Girard, « Aries Philippe L'homme devant la mort », *Population*, 33-2 (1978), p. 471-472, URL : https://www.persee.fr/doc/pop_0032-4663_1978_num_33_2_16750 (visité le 21/09/2020)

[...] l'instruction a demandé de communiquer à 75 ans les minutes demandées par les chercheurs universitaires pour des recherches conduites selon un protocole d'histoire quantitative, considérant que, dans ce type de recherches, le risque d'indiscrétions apparaît limité, puisque l'intérêt des chercheurs porte sur l'analyse sérielle de faits de société et non sur l'étude de cas particuliers.⁵⁴

Les archives notariales sont des sources connues des historiens. Les humanités numériques et les outils du TAL ouvrent la perspective d'un « nouveau regard par les historiens »⁵⁵ sur les archives notariales.

Dès lors, l'extraction d'informations tels que le prix des actes pratiqués, la taxation de certains actes ou encore le capital des entreprises, autorise, en combinaison avec des outils informatiques de visualisations statistiques, de mener ou de compléter des études sur l'évolution de la fiscalité, de la masse successorale⁵⁶, et le poids financier des entreprises pour évaluer les périodes de creux dans une activité économique. Mais nous ne présentons ici que quelques exemples des nombreuses approches permises par cette technologie.

L'analyse des réseaux (sociaux), approche issue de la sociologie, peut permettre de visualiser les interactions sociales entre des personnes. Cette technique déjà éprouvée en histoire médiévale⁵⁷ a permis d'augmenter la compréhension l'analyse des réseaux marchands, monastiques ou d'amitiés. Il en est de même pour les généalogistes pour qui l'étude des lignages, sous la forme de réseaux avec des liens de type orientés (« être le parent de ») et des liens non-orientés (« être marié à »), constitue une approche méthodologique valable⁵⁸.

54. M.F. Limon-Bonnet et G. Étienne, *Les archives notariales : manuel pratique et juridique...*

55. M.F. Limon-Bonnet, J.F. Moufflet et G. Piraino, « L'innovation numérique : un cercle vertueux pour l'archivistique »..., pp.265

56. La masse successorale est souvent définie comme la part des actifs (ensemble des biens du défunt) dont on déduit un passif (ensemble des dettes).

57. Laurent Jégou, *Potentialités de l'analyse-réseau en histoire médiévale*, COL&MON, 2017, URL : <https://colemon.hypotheses.org/102> (visité le 14/09/2020)

58. Laurent Beauguitte, « L'analyse de réseaux en sciences sociales et en histoire », dans *Le réseau. Usages d'une notion polysémique en sciences humaines et sociales*, Presses Universitaires de Louvain, 2016, p. 9-24, URL : <https://halshs.archives-ouvertes.fr/halshs-01476090> (visité le 14/09/2020), pp.12.

2.3 Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel

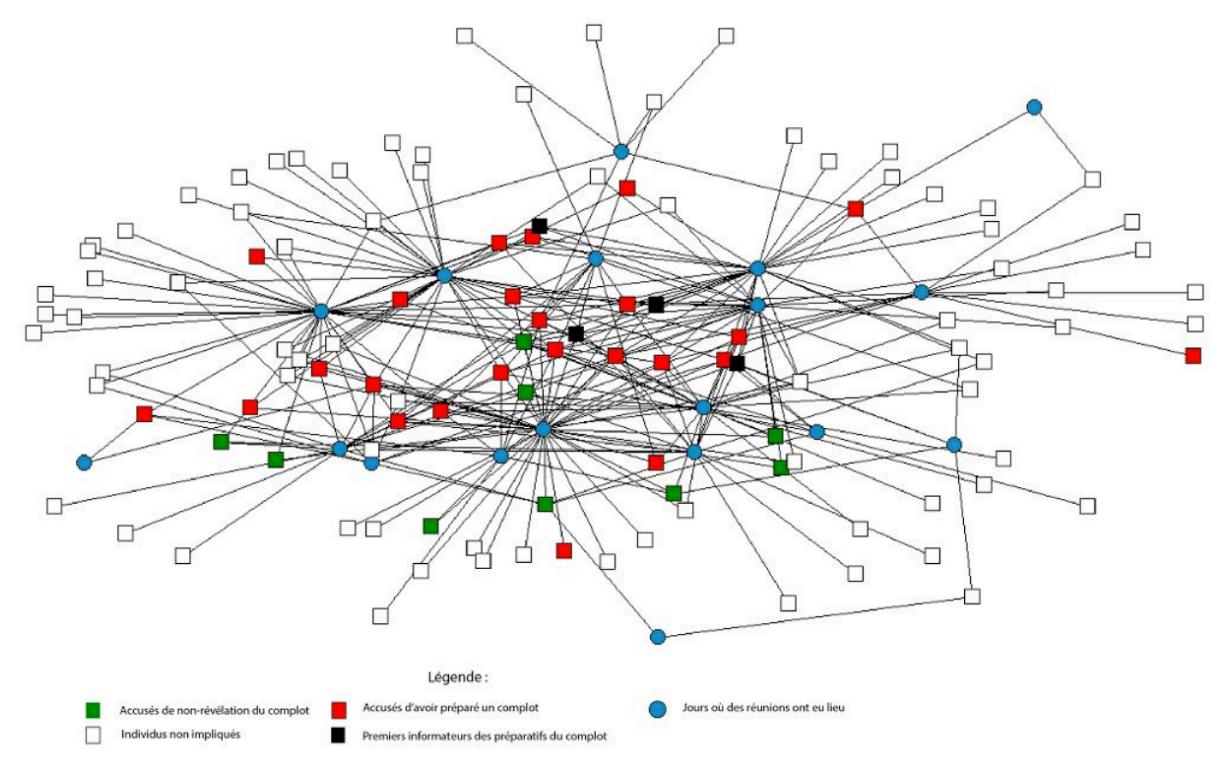


FIGURE 2.14 – Un exemple de représentation en réseaux sous la forme d'un graphe entre acteurs du complot (suivant les charges retenues contre eux) et des dates dans le cadre d'une étude du complot 19 août 1820 contre Louis XVIII. ©FARAUT (Vivien), *Les outils de représentation graphique de l'espace relationnel face au secret : le cas des conspirateurs du 19 août 1820*, Les Cahiers de Framespa. Nouveaux champs de l'histoire sociale, 2015, URL : <http://journals.openedition.org/framespa/3233> (visité le 15/09/2020)

Cependant, les visualisations en réseaux sont concomitantes de la propreté des données et de leur quantité sélectionnées en amont du projet. Un trop grand nombre de données peut faire croître le nombre de relations, ce qui peut rapidement rendre illisible le graphe de données⁵⁹.

Ces visualisations ne se limitent pas aux relations entre personnes, mais s'intéressent également aux données géographiques. L'usage d'outils développés dans le cadre des humanités numériques comme, par exemple, Palladio⁶⁰, créé par l'Université de Standford, permettent de cartographier ou de visualiser de vastes ensembles de données géolocalisées.

Pour Lectaurep, les outils de visualisations en réseaux constituent une plus-value pour l'évaluation de la répartition topographique et géographique de la clientèle ou le rayonnement de l'activité d'un notaire sur des cartes de l'époque.

59. *Ibid.*, pp.15-16.

60. Standford University, *Palladio*, URL : <https://hdlab.stanford.edu/palladio/>

Deuxième partie

**Représenter et homogénéiser dans
un format pivot XML TEI les
métadonnées Lectaurep**

Nous venons de présenter les spécificités de la chaîne de traitement de Lectaurep qui fait circuler des ensembles de données structurées complexes qui sont manipulés et ne communiquent actuellement pas forcément entre-eux. Le premier objectif du DMC est de pouvoir récupérer les métadonnées de ces traitements spécifiques dans *eScriptorium* et les associer à leurs informations déjà présentes en salle des inventaires virtuelles (SIV) pour enrichir et valoriser leurs fonds.

La première partie du stage comportait une réflexion sur la gestion et la création des métadonnées dans le cadre du projet Lectaurep et de la possibilité de les unifier au sein d'un même fichier. Cette mission s'est présentée sous la forme de questions : quelles métadonnées sont susceptibles d'être intégrées à la plate-forme *eScriptorium* lors de l'import des images ? Quelles métadonnées extraire une fois les traitements dans cette même plate-forme effectués ? Par ailleurs, quelles solutions pour modéliser et rendre interopérables les métadonnées repérées ? Enfin, sur quel(s) référentiel(s) s'appuyer ?⁶¹

Les réponses à apporter à ces questions ont nécessité plusieurs étapes de réflexion et la mise en place d'une stratégie dans l'application des technologies.

Dans un premier temps, une présentation de l'ensemble des métadonnées, de leurs référentiels associés ainsi que de leur circulation en terme de flux entrants et sortants dans la plate-forme *eScriptorium*, s'est avérée cruciale. En concertation avec l'ensemble des acteurs du projet, nous avons pu repérer un certain nombre de métadonnées à associer aux images lors des étapes d'import et d'export dans *eScriptorium*.

Dans un second temps, nous avons posé les avantages d'agréger ces données dans un fichier pivot XML répondant à la spécification TEI (*Text Encoding Initiative*). Ce fichier doit permettre, à terme, d'échanger des ressources entre différents systèmes d'informations de Lectaurep. Il fut nécessaire de mettre en avant les atouts du modèle de la TEI, dans un contexte archivistique, en vue de la continuation du projet Lectaurep sur le long terme.

Une fois les définitions et les questionnements posés, nous avons pu préparer un environnement propice au développement « agile » de ce format pivot. L'intention était de garantir sa future évolution et d'expérimenter une première formalisation dans un fichier XML TEI (créé *ex nihilo* pour l'occasion), avec un petit choix de métadonnées. Nous résumons ici ces différentes phases logiques.

61. La première mission a été présentée sous la forme d'une *issue* GitLab, URL : <https://gitlab.inria.fr/almanach/lectaurep/documentation/-/issues/1> [à titre informatif, un accès peut être requis]

Chapitre 3

Enjeux et analyse des problématiques liées aux données Lectaurep et au format pivot XML-TEI

3.1 De l'importance de l'interopérabilité des données

3.1.1 Définitions et objectifs de Lectaurep

3.1.1.1 Rappels généraux : données et formats

Actuellement, la donnée (*data*) est partout, cependant l'usage de ce terme peut être trompeur et souvent mal compris. Elle n'est pas simplement quelque chose qui existe en soi, mais peut faire également l'objet d'une construction. Le terme de « donnée » renvoie alors à des contextes et des réalités différentes.

D'après le CNRTL, la **donnée** est ce qui est communément admis et qui sert de base à une recherche ou un à raisonnement¹. D'après cette définition c'est donc une information codifiée, manipulable, transmissible et figée comme une mesure ou un fait observable. Ainsi la donnée brute ou primaire correspond à l'information tel qu'existe, par exemple, un acte de notaire qu'une historienne ou un historien s'apprête à étudier.

En informatique, ces données manipulées sur un ordinateur sont de deux types. Dans un premier temps, les **données non structurées**. Ce sont celles qui existent sur un support informatique, mais qui n'ont pas fait l'objet d'une modélisation conceptuelle

1. « donnée », CNRTL (Centre national des ressources textuelles et lexicales), URL : <https://www.cnrtl.fr/definition/academie8/donn%C3%A9e>

et ne sont pas stockées selon un format qui correspond à l'implémentation technique du modèle, comme par exemple du texte brut où les caractères sont représentés selon le standard d'encodage UTF-8. Au contraire, les **données structurées** ont fait l'objet d'une conceptualisation durant laquelle on a pu déterminer des propriétés et des champs de valeurs possibles ; cette conceptualisation se traduit dans un format technique (l'information est codée). Une métadonnée est également une donnée structurée, dans le sens où elle est une donnée sur la donnée servant à décrire les caractéristiques d'une donnée brute en vue de faciliter son repérage, sa compréhension, sa gestion, son usage ou sa préservation selon les cas. Nous parlerons en l'occurrence, de données structurées principalement.

Les données sont généralement représentées dans plusieurs types de formats au sein d'un **système d'information**² (SI) comme par exemple un tableau à double entrée, une base de données relationnelles ou un document XML. Le monde documentaire utilise principalement les formats basés sur les langages à balises reconnaissables grâce à leurs chevrons ouvrants et fermants (< >). Les plus connus sont ceux qui descendent du langage SGML³ (*Standard Generalized Markup Language*) et sont maintenus par le W3C⁴ (*World Wide Web Consortium*). Par exemple, le HTML⁵ (*Hyper Text Markup language*) qui permet de rendre compte d'une mise en page ou d'une typographie particulière d'un ensemble de données (texte stylisé ou *fancy text*) pour les exposer le *web*.

Omniprésent dans l'informatique documentaire, le balisage XML⁶ (*Extensible Markup Language*) permet de décrire la logique et la sémantique des données.

Un document XML respecte généralement des règles de syntaxe précises. Parmi les règles qui définissent un document XML « bien formé » ou *well-formed* : les principes

2. Un système d'information (SI) est un ensemble organisé de ressources (matérielles et logicielles) qui permettent de collecter, stocker, traiter et distribuer les données/métadonnées.

3. Le **SGML** ou *Standard Generalized Markup Language* est un langage de description à balises. Apparu dans les années 1980, il a rendu possible l'exploitation des premières DTD (*Document Type Definition*) qui est un schéma de déclaration des éléments utilisés selon des règles. Il est aujourd'hui abandonné au profit de ses descendants HTML, XHTML et XML apparus au moment de l'invention du *web* dans les années 1990.

4. Le W3C (*World Wide Web Consortium*) est le principal organisme de standardisation des langages du *web*, URL : <https://www.w3.org>

5. Le **HTML** ou *Hyper Text Markup language* est le langage de la page *web* par excellence. Il sert à structurer sémantiquement du contenu et à rendre compte de son apparence. Cependant la mise en page du HTML est souvent réalisée avec le langage de programmation JavaScript (apparence dynamique) et du CSS (*Cascading Style Sheets* pour l'apparence statique et dynamique de la page *web*).

6. Le **XML** ou *Extensible Markup Language* est un langage de balisage extensible, héritier du SGML. Au même titre que son frère HTML, il est reconnaissable par l'usage de chevrons ouvrants et fermants : < >. Sa syntaxe est dite « extensible » car on peut définir ses propres noms de balises (grammaire). Les objectifs du XML sont l'encodage de ressources textuelles via son système de balises, l'échange de contenus complexes pouvant être représentés sous la forme d'arbres entre systèmes d'informations hétérogènes (interopérabilité). Un document XML pour être « valide » doit respecter un schéma défini ou non par les utilisateurs (par exemple, une DTD ou un schéma RelaxNG) et posséder une syntaxe « conforme ». Ce qui est en fait un langage strict en dépit de son apparence d'« extensibilité ».

3.1 De l'importance de l'interopérabilité des données

d'arborescences par imbrication d'éléments (l'élément enfant hérite des propriétés et attributs d'un élément parent) et non de recouvrement ; à chaque balise doit correspondre une balise de fin ; il existe un seul élément racine (*root element*) ; un élément ne peut posséder deux attributs du même nom. Les principes du langage XML sont résumés dans l'exemple ci-dessous :

```
1 <?xml version="1.0" encoding="utf-8"?> <!-- déclaration XML obligatoire -->
2 <elementParent couleur='rouge' position='milieuDroitePage'>
3   <unElementEnfant id='1'>du texte</unElementEnfant>
4   <unElementEnfant id='2'>encore du texte</unElementEnfant>
5 </elementParent>
```

Le format XML présente des avantages conséquents dans la représentation des données, il permet entre autre :

- une bonne lisibilité par l'humain et la machine ;
- une facilité d'intégration dans des plate-formes et logiciels ;
- une migration vers d'autres formats ;
- l'échange de données.

Toutefois, le format XML s'appuie généralement sur des spécifications qui décrivent des catégories d'objets en particulier. Ainsi une bibliothèque et un centre d'archives ne verront pas de la même manière une archive et un livre en XML.

Les spécifications ou standards de représentation des données et des métadonnées répondent généralement à des normes techniques d'échange⁷ de données et de métadonnées selon le contexte d'utilisation.

Les normes ISAD(G)⁸ et ISAAR(CPF)⁹ sont en vigueur dans le domaine archivistique pour décrire respectivement des fonds d'archives et des producteurs.

7. Une **norme technique** est un référentiel publié par un organisme de normalisation officiellement agréé par l'État via une organisation nationale (comme AFNOR - Association française de normalisation) ou internationale (comme ISO - Organisation internationale de normalisation). Le but étant d'harmoniser les pratiques d'un secteur d'activité suivant un cadre commun et de garantir un niveau d'ordre optimal.

8. **ISAD(G)** (*International Standard Archival Description-General*) ou Norme générale et internationale de description archivistique créé en 1994 sous l'impulsion du Conseil International des Archives (CIA, ICA en anglais) énonce les grands objectifs et principes de la descriptions archivistique et fournit une liste d'éléments de description que l'on peut employer dans un instrument de recherche, avec la définition de ces éléments et des exemples d'utilisation.

9. **ISAAR(CPF)** (*International Standard Archival Authority Record for Corporate Bodies, Persons and Families*) ou Norme internationale sur les notices d'autorité archivistiques relatives aux collectivités, aux personnes et aux familles, est créé en 1995 par le Conseil International des Archives (ICA). La norme énonce les lignes directrices pour la description d'entités (collectivités, personnes, familles) associées à la production et à la gestion d'archives.

Dans le monde des bibliothèques, le catalogage repose également sur un ensemble de normes comme l'ISBD¹⁰ ou le *Dublin Core* (norme ISO 15836) pour décrire des ressources bibliographiques.

Dès lors il existe des implémentations de ces normes en XML. La norme ISAD(G) se trouve décliné en standard XML EAD¹¹ et ISAAR(CPF) en standard XML EAC-CPF¹².

Ces standards d'encodage sont généralement disponibles sous la forme d'un schéma de validation XML de type Relax NG¹³ ou DTD¹⁴ directement épinglé au document XML pour vérifier la strict validé de ce dernier. Si ces standards facilitent l'échange de certains type de données, ces dernières peuvent-elles communiquer entre elles ?

Nous verrons en détail dans la section 3.1.2, les formats et spécifications propres aux données et métadonnées qui nous ont particulièrement intéressés dans le cadre du projet.

10. **ISBD** (*International Standard Bibliographic Description*) ou Description bibliographique internationale normalisée est une norme publié en 1971 et définie par l'IFLA (Fédération internationale des associations de bibliothécaires et d'institutions) pour la description du catalogage.

11. **EAD** (*Encoded Archival Description*) ou description archivistique encodée, créé en 1993 et maintenue par la Bibliothèque du Congrès. S'appuyant sur la norme ISAD(G), l'EAD est utilisé pour décrire des fonds d'archives, des collections de manuscrits ou des collections hiérarchisés d'objets (photographies, numérisations, microfilms etc.), mais surtout pour encoder des instruments de recherche. L'ensemble des éléments XML sont présentés sur le site officiel : <https://www.loc.gov/ead/> et en français : <https://www.ead-bibliotheque.fr>

12. **EAC-CPF** (*Encoded Archival Context - Corporate Bodies, Persons and Families*) s'appuie sur la norme définie par ISAAR(CPF) pour la rédaction des fiches de producteurs d'archives. C'est un schéma maintenu par la Bibliothèque d'État de Berlin et la SAA (*Society of American Archivists*). Les éléments de l'EAC sont disponibles : <https://eac.staatsbibliothek-berlin.de>

13. **Relax NG** (*Regular Language for XML Next Generation*) est un langage de description de document XML permettant de définir les différentes contraintes qui déterminent l'imbrication des éléments ou la place des attributs dans le document pour passer l'étape de validation. C'est une puissante alternative au langage XML Schema.

14. **DTD** (*Document Type Definition*) ou définition de type de document, est un document XML qui défini un modèle pour des fichiers XML et réglemente l'imbrication des balises, les noms d'éléments autorisés, la façon dont les attributs sont rattachés aux éléments. C'est un schéma XML moins sophistiqué que le Relax NG et XML Schema.

3.1 De l'importance de l'interopérabilité des données

3.1.1.2 La question de l'interopérabilité dans Lectaurep

Derrière ces définitions très généralistes, reposent des enjeux essentiels et concrets pour le projet Lectaurep et les utilisatrices et les utilisateurs de la plate-forme *eScriptorium*.

Les répertoires de notaires numérisés qui sont importés dans la plate-forme *eScriptorium* ne sont actuellement accompagnés d'aucune, sinon de très peu de métadonnées. Dès lors une annotatrice ou un annotateur qui travaille sur la transcription ou la segmentation d'une image, ne sera pas en capacité d'avoir des informations précises issues des instruments de recherche (producteur d'un document d'archives, cotes, informations techniques sur l'image). Un chercheur ou une chercheuse peut souhaiter des informations précises sur le fonds d'où est issue une transcription de répertoire.

Lectaurep en tant que projet d'analyse et de reconnaissance de documents (ARD) est confronté au problème de la représentation des données pour couvrir des besoins utilisateurs diversifiés. Ainsi comme le souligne des chercheurs du LORIA (laboratoire lorrain de recherche en informatique et ses applications) et du laboratoire CNRS/ATILF de Nancy (Analyse et Traitement Informatique de la Langue Française) :

Il existe pour ainsi dire autant de formats de représentation des données que de systèmes de stockage ou de reconnaissance de document. Cette grande diversité est un frein certain à l'échange des données d'un environnement à un autre, d'une plate-forme à une autre et rend souvent les sorties de certains systèmes utilisables uniquement par eux-mêmes.¹⁵

Le premier objectif pour Lectaurep est donc celui de l'**interopérabilité**¹⁶ des données et des métadonnées dans *eScriptorium*. Soit la capacité de faire communiquer les données et métadonnées entre-elles, de les relier. Les données et métadonnées susceptibles d'être importées dans *eScriptorium* suivent pour l'heure des modèles différents.

15. Abdel Belaïd, Ingrid Falk et Yves Rangoni, « Représentation des données en XML pour l'analyse d'images de documents », *Conférence Internationale sur l'Ecrit et le Document* (, 2007), URL : <http://lodel.irevues.inist.fr/cide/index.php?id=147> (visité le 07/09/2020), pp.1-2

16. On parle généralement du principe ou de l'objectif FAIR (Facile à trouver, Accessible, Interopérable, Réutilisable) dans une plate-forme pour évoquer le fait que les données doivent être échangeables et identifiables pour permettre des bonnes performances.

La plate-forme *eScriptorium* prend en charge des formats d’images standards (.jpg, .jpeg, .png et .tif) via une fonctionnalité de type « Glisser/Déposer » (*Drag & Drop*) ou avec l’URI¹⁷ du manifeste IIIF. Les transcriptions sont chargées dans les formats XML ALTO¹⁸ ou en XML PAGE¹⁹ pour les utilisatrices ou les utilisateurs qui souhaiteraient reprendre leurs travaux de transcription en cours. Lors de l’export, on peut récupérer le résultat de sa transcription (pour la sauvegarde notamment) au format XML ALTO (version 4.0), XML PAGE ou en texte brut.

Dès lors, on retrouve un format de fichier par type de données, certains formats étant mieux adaptés pour contenir et décrire certains types de données que d’autres. En revanche, cela présente le désavantage, pour les utilisatrices ou les utilisateurs, de ne pas avoir accès à l’ensemble des données et des métadonnées lors de l’import/export de leurs travaux dans la plate-forme.

Idéalement, lors de l’import et de l’export les utilisateurs devraient pouvoir récupérer en une fois dans *eScriptorium* :

- l’image de la numérisation du répertoire de notaires, sous la forme d’une URI IIIF ou encore d’un chemin vers une ressource stockée localement ;
- le résultat de sa transcription (vérité terrain ou résultat HTR) ;
- les métadonnées sur l’archive (producteurs, date, étude de notaire etc.), provenant des instruments de recherche du DMC ;
- l’historique de ses traitements (binarisation, segmentation, transcription etc.).

17. **URI** (*Uniform Resource Identifier*) ou identifiant uniforme de ressource est une chaîne de caractère identifiant une ressource dans un réseau répondant à une norme du W3C pour le *web* (Cf. RFC 3986). Une URL (*Uniform Resource Locator*) comme par exemple « <http://www.lectaurep.fr/> » est un type d’URI qui identifie la ressource (la page de lectaurep), qui implique la représentation de la ressource en HTML obtenue via le protocole HTTP (*Hypertext Transfer Protocol*, principal protocole de communication client-serveur dans le *web*).

18. **ALTO** (*Analysed Layout and Text Object*) est un standard XML rendant compte de la mise en page physique et de la structure logique d’un texte transcrit par reconnaissance optique de caractère (OCR/HTR). On y retrouve les coordonnées de segmentation (*baseline*, polygones etc.), des éléments de forme (type de police etc.), ou encore des métriques (taux de confiance de reconnaissance). Le schéma de ce standard est maintenu par la Bibliothèque du Congrès et la BnF. <https://www.loc.gov/standards/alto/>

19. **PAGE XML** est le principal format de données utilisé par le logiciel Transkribus, il agrège l’ensemble des données relatives à l’image transcrise, le texte transcrit, les coordonnées de zones de texte, les corrections effectuées etc. URL : <https://www.primaresearch.org/schema/PAGE/gts/pagecontent/2016-07-15/pagecontent.xsd>

3.1 De l'importance de l'interopérabilité des données

3.1.1.3 Une solution, le format pivot XML

La Figure D.1 en Annexes représente le modèle hypothétique envisagé par Lectaurep pour l'utilisation du format pivot et de ses usages à terme dans *eScriptorium*.

Avant d'aborder la TEI comme modèle de données, le format pivot XML est la solution qui a été envisagé pour permettre une intéropérabilité des spécifications des données Lectaurep dans *eScriptorium* durant mon stage. Il représente un bon point de départ pour traiter et agréger des flux entrants et sortants de données et de métadonnées dans une application. Cependant, pour être opérationnel et intégrable, le format pivot doit répondre aux critères²⁰ de :

- « délimitation » : simple à utiliser pour n'importe quel type d'applications, les données et les métadonnées utilisées doivent être identifiables dans ce format ;
- « soutenance » : intégrer à un environnement technique permettant son évolution (modification), son implémentation facile et sa documentation ;
- « spécification » : chaque projet répond à des caractéristiques originales qui lui sont propres, cependant l'appui sur un schéma déjà existant et ouvert (modulable) permet de créer une ligne directrice, pour un développement rapide.

Nous verrons que si le format pivot peut sembler indispensable pour faire converger des données de différents SI dans Lectaurep, le choix du modèle de représentation des données Lectaurep dans ce format pivot s'avère en revanche moins évident à définir (Cf. parties 3.2.2 et 3.2.3).

3.1.2 « Circonscrire un monde », un focus sur les données de Lectaurep

Avant de concevoir la mise en place d'un fichier pivot XML, il faut comprendre quels sont les types de données que l'on va y intégrer. Nous pouvons décrire cette étape comme « circonscrire le monde »²¹ des données et des métadonnées de Lectaurep. Ainsi j'ai effectué un premier travail de repérage durant le stage sur l'ensemble des données et métadonnées circulant dans les SI rattachés à Lectaurep. Ils sont donc susceptibles d'être récupérés lors des phases d'import ou d'export des images dans *eScriptorium*, dans le fichier pivot XML. La figure 3.1 montrent ces données, leur circulation et leurs relations en terme de SI. Nous avons cherché à brosser le tableau des métadonnées et des données

20. Ces critères d'évaluation du format pivot sont adaptés de l'article Stéphane Crozat, « Standardisation des formats documentaires pour les chaînes éditoriales d'UNIT : un schéma pivot », *TICE* (, 2006), URL : <https://stph.crozat.fr/res/crozat06tice.pdf> (visité le 07/09/2020)

21. Gautier Poupeau, *Visite guidée au pays de la donnée - Du modèle conceptuel au modèle physique*, cours, 2019, URL : <https://fr.slideshare.net/lespetitescases/visite-guide-au-pays-de-la-donne-du-moble-conceptuel-au-moble-physique> (visité le 13/08/2020)

de Lectaurep de la manière suivante :

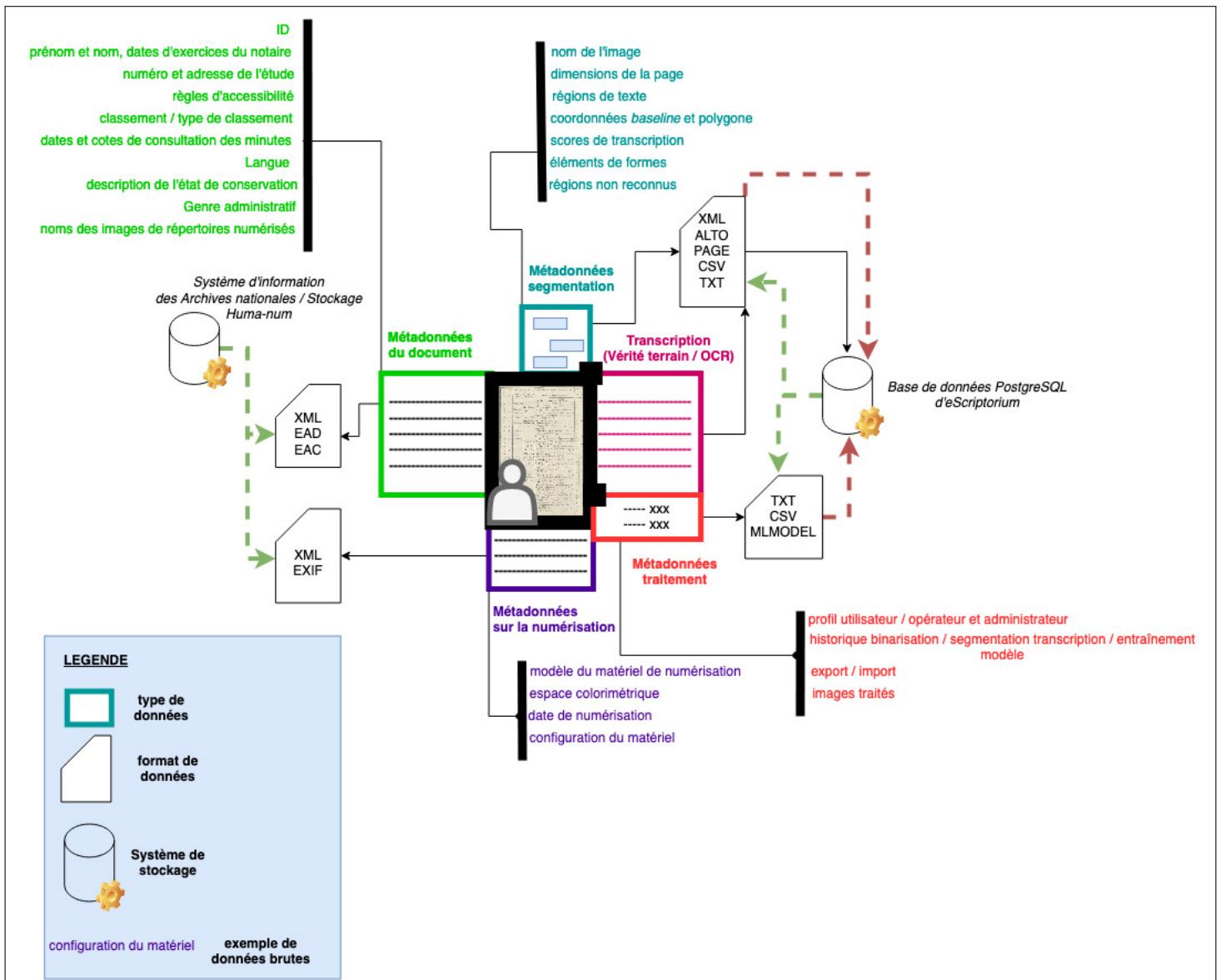


FIGURE 3.1 – Modélisation conceptuel des données, des formats et des systèmes d'informations du projet Lectaurep ©L. Terriel, 2020, Diagrams.net

3.1 De l'importance de l'interopérabilité des données

Informations relatives aux répertoires de notaires en tant que documents d'archive - Il s'agit des informations contenues dans les instruments de recherche²² (IR) en XML EAD et les notices producteurs (NP) en XML EAC-CPF²³. Ils sont récupérables dans la Salle des inventaires virtuels (SIV) dépendant du SI des AN²⁴ et parfois sur l'espace *ShareDocs* dédié au projet.

Le DMC disposent de deux types d'IR par notaire tel que :

- un instrument de recherche sous la forme « Minutes et répertoires du notaire *M^eN* » : il s'agit d'un inventaire des actes, accompagné de descriptions détaillées de chacune des minutes de notaires et des répertoires eux-même ;
- un instrument de recherche sous la forme « Images des répertoires du notaire *M^eN* » : il s'agit d'un inventaire spécifique les images numériques du répertoire d'un notaire donné.

Ainsi que deux types de NP par notaire tel que :

- une notice producteur spécifique à l'étude du notaire ;
- une notice producteur spécifique au notaire lui-même.

Ces fichiers contiennent généralement des métadonnées administratives pour la gestion (identification des archives, provenance et contexte de production de ces archives, intégrité, droits, etc.) et des métadonnées descriptives telles que les nom et prénom du notaire, le numéro de l'étude, l'adresse, le type de répertoires ou autre.

Ces métadonnées sont reliées à la donnée brute, le répertoire en tant qu'archive matérielle, et à la copie numérique de celui-ci, son image. Dès lors, il existe entre ces différents IR et les NP des liens et des informations redondantes. Il est alors possible de faire en sorte de les croiser dans le fichier pivot XML, à l'image du numéro de l'étude, du nom du notaire, la cote du répertoire, les dates extrêmes d'exercice du notaire par exemple.

22. Un **instrument de recherche** est un outil papier ou informatisé énumérant ou décrivant un ensemble de documents d'archives de manière à les faire connaître aux lecteurs, définition issue de Lydiane Gueit-Montchal (Dir.), *Abrégé d'archivistique. Principes et pratiques du métier d'archiviste 4^e édition revue et augmentée*, AAF, 2020

23. Pour consulter des exemples de ces fichiers Cf. Annexes, [/B-Format_pivot_XML_TEI_Lectaurep/Generator_Lectaurep2TEI/Sets_test_Legay/Data_xml_ead_eac/](#)

24. Salle des inventaires virtuels des Archives nationales, URL : <https://www.siv.archives-nationales.culture.gouv.fr/siv/cms/content/display.action?uuid=AccueilRootUuid&onglet=1>

Informations techniques relatives aux images numérisées de répertoires de notaires - Ces métadonnées sont généralement contenues dans les images des répertoires numérisés. Les images sont pour l'instant disponibles sur l'espace *ShareDocs* dédié au projet. À terme, Lectaurep à pour ambition de disposer d'un serveur IIIF dédié au stockage des images et à la récupération de celles-ci, notamment par l'intermédiaire d'URI stables. L'avantage de ce stockage repose sur le fait qu'*eScriptorium* peut consommer des images IIIF à partir de ces URI²⁵.

Les métadonnées techniques des images, qui ont intéressé le DMC durant mon stage, correspondent aux informations sur la numérisation des images et sont exposées au format EXIF²⁶ (*Exchangeable image file format*). Pour y avoir accès, on utilise un logiciel de traitement d'images de type *GIMP* ou *XnView* ou par le biais d'un script Python utilisant le package *Pillow*²⁷ ou *PyExifTool*²⁸ (Cf. Figure 3.2).

```
1 EXIF:ImageDescription => 74 - Main frame
2 EXIF:Make => i2S, Corp.
3 EXIF:Model => SupraScanII [SN: 283910] - Cam7600RGB [SN: 283910]
4 EXIF:Orientation => 1
5 EXIF:XResolution => 300
6 EXIF:YResolution => 300
7 EXIF:ResolutionUnit => 2
8 EXIF:ModifyDate => 2015:09:07 09:45:39
9 EXIF:YCbCrPositioning => 1
10 EXIF:ExifVersion => 0230
11 EXIF:CreateDate => 2015:09:07 09:45:39
12 EXIF:ComponentsConfiguration => 1 2 3 0
13 EXIF:FlashpixVersion => 0100
14 EXIF:ColorSpace => 65535
```

FIGURE 3.2 – Un exemple de métadonnées EXIF extraites d'une image de répertoire numérisée et affichées en sortie d'un script Python (Cf. Annexes, Figure F.2) ©L. Terriel, 2020, *Pycharm*

Enfin , il est possible d'exposer les données répondant à la spécification EXIF dans un format XML à condition d'utiliser le bon espace de nom sur les balises et le bon schéma XML associé.

25. Actuellement, *eScriptorium* semble faire une copie locale de l'image récupérée depuis IIIF ; c'est-à-dire la charge et la duplique. Ce qui n'est effectivement pas souhaité sur le long terme.

26. **EXIF** ou *Exchangeable image file format* est une spécification de format de fichier pour les images utilisés par les appareils photographiques numériques ou les appareils de numérisation. Cette spécification repose sur des formats existants tels que le JPEG, JPG, TIFF (mais pas le JPEG 2000 ni le PNG. Pour accéder aux principaux champs EXIF et à leurs descriptions : <https://exiftool.org/TagNames/EXIF.html>

27. *pillow* 7.2.0, URL : <https://pillow.readthedocs.io/en/stable/>

28. *PyExifTool* 0.1, URL : <http://smarnach.github.io/pyexiftool/>

3.1 De l'importance de l'interopérabilité des données

Informations liées à la transcription des vérités terrains et HTR des répertoires de notaires - Les agents des AN rattachés à Lectaurep réalisent actuellement dans la plate-forme *eScriptorium* des transcriptions à partir des images de répertoires dans le but d'éditer des vérités terrains afin d'entraîner des modèles de segmentation. Ces données peuvent être importées ou exportées dans des fichiers XML ALTO, XML PAGE ou encore en texte brut. À terme, la transcription réalisée automatiquement par le modèle HTR dans *eScriptorium* sera exportée dans ces types de fichiers. Pour prendre le cas des XML ALTO²⁹ - format privilégié pour l'export des transcriptions de vérité terrain - on y retrouve parfois des informations relatives à la mise en page physique, à l'emplacement d'une zone texte et d'une ligne de base - *baseline* - sous la forme de coordonnées qui situent les zones de textes - paragraphe et polygones - et la ligne de base du texte. On retrouve également le texte transcrit par l'utilisateur ou l'utilisatrice d'*eScriptorium*.

Informations liées à l'historique des traitements dans la plate-forme eScriptorium - *eScriptorium* conserve la trace des transactions : informations sur le compte de l'usager, historique des traitements (segmentation, binarisation, transcription, etc.) et des opérations d'import et export. Ces données sont stockées dans le *back-office* de l'application, dans une base de données SQL³⁰ de type *PostgreSQL*³¹. Certaines de ces données³² devront à terme être intégrées dans le fichier pivot XML de Lectaurep à l'import et à l'export des images dans *eScriptorium*.

Nous avons déjà souligné, la notion selon laquelle l'encodage de ces données en XML constituait un dénominateur commun, exception faite pour la dernière catégorie évoquée précédemment, accessible par l'intermédiaire du langage SQL. Les parties qui vont suivre, proposent de mettre en avant les avantages qu'offre une modélisation TEI des données Lectaurep. Durant ce stage, nous avons choisi de nous concentrer sur un choix restreint de données et de métadonnées. Elles proviennent des catégories évoquées plus haut, et permettent de constituer un premier modèle de fichier pivot XML-TEI, ce sur quoi nous reviendrons également par la suite.

29. Pour consulter des exemples de ces fichiers Cf. Annexes, /B-Format_pivot_XML_TEI_Lectaurep/Generator_Lectaurep2TEI/Sets_test_Legay/Data_xml_alto/

30. SQL (*Structured Query Language*) ou langage de requête structuré, est un langage informatique servant à exploiter des bases de données relationnelles. Il permet principalement de rechercher (SELECT), d'ajouter (INSERT), de supprimer (DELETE), ou de modifier (UPDATE) des données dans la base constituée.

31. PostgreSQL est un type de système de gestion de base de données relationnelle et objet (SGB-DRO).

32. On peut avoir un aperçu des types et des migrations de données opérées lors des transactions dans l'interface en parcourant le code-source en Python de l'application *eScriptorium*, <https://gitlab.inria.fr/scripta/escriptorium/-/tree/master/app/apps>

3.2 Le format pivot XML TEI, un choix réaliste ?

3.2.1 La TEI pour annoter les données et les métadonnées de Lectaurep

La *Text Encoding Initiative*³³ (TEI) est sans doute l'un des plus importants projets d'application de l'informatique au domaine des sciences humaines et sociales. Dans « 40 ans de relations entre les sciences humaines et informatique »³⁴, Lou Burnard, chercheur de l'Université d'Oxford et cofondateur de la TEI, fait remonter l'apparition du projet TEI au deuxième âge des humanités numériques situé au début des années 1980. Après une période qu'il intitule « *Litterary and linguistic Computing* » marqué par les techniques statistiques de l'histoire quantitative, la TEI émerge dans la période des « *Humanities Computing* ». Le chercheur s'éloigne alors du paradigme quantitatif (sans réellement le quitter³⁵), pour encoder et structurer des informations qu'il paraît utile de conserver et d'exploiter.

La TEI, créée en 1987, vise à standardiser les pratiques d'encodage et de structuration des textes chez les chercheuses et les chercheurs. Mise en pratique dans un premier temps dans le cadre du langage SGML, elle devient ensuite une norme d'encodage pour le langage XML, applicable, en principe, à n'importe quelle source textuelle numérisée. L'outil est structuré autour d'une communauté pluridisciplinaire, composée d'historien(ne)s, de linguistes, de philologues ou encore d'archéologues. Ces professions utilisent et adaptent cette norme selon leurs besoins tout en suivant les recommandations de l'un des 21 modules d'éléments décrits dans les *guidelines TEI P5*³⁶.

Il s'agit d'une avancée considérable, évitant la sédimentation ou la « babélistation » numérique, où chacun proposerait son propre langage d'encodage suivant sa propre théorie. Ce « métalangage » commun offre un cadre de travail et une structure assez souple pour pouvoir s'adapter à chaque projet ou discipline.

Ce dispositif commun permet en particulier, une intéropérabilité minimale entre les données, objectif apprécié et recherché dans le projet Lectaurep. Au travers de la norme TEI, il sera possible d'encoder les informations des différents SI de l'éco-système

33. Pour de plus de détails sur l'histoire, le fonctionnement et les usages de la TEI consulter Lou Burnard et Marjorie Burghart, *Qu'est-ce que la Text Encoding Initiative?*, OpenEdition Press, 2015, URL : <http://books.openedition.org/oep/1237>

34. Lou Burnard, « Du *literary and linguistic computing* aux *digital humanities* : retour sur 40 ans de relations entre sciences humaines et informatiques », in Pierre Mounier (dir.), *Read/write book 2 : une introduction aux humanités numériques*, OpenEdition Press, 2012, URL : <https://books.openedition.org/oep/226?lang=fr>

35. En 1983, la revue *Histoire & Mesure* paraît en France.

36. Les *guidelines TEI P5*, manuels, tutoriels, outils pour l'encodage de texte sont accessibles à l'adresse : <https://tei-c.org/guidelines/P5/>

3.2 Le format pivot XML TEI, un choix réaliste ?

Lectaurep (présenté dans la partie 3.1.2) dans un format qui permet leur structuration, leur alignement (au moyen de liens entre celles-ci), leur diffusion (vers d'autres formats de données ou d'autres programmes, par exemple) et leur archivage à long-terme, mais aussi, comme nous l'évoquions en section 2.3.2, d'annoter des informations plus spécifiques comme les entités nommées issues des traitements du TAL.

3.2.2 Faire converger les standards de données vers un fichier pivot XML-TEI : confronter des visions opposées sur le document...

Lors d'un colloque organisé, en 2019, aux Archives diplomatiques de La Courneuve, le chercheur en humanités numériques Seth Van Hooland rappelait en substance que « les standards de métadonnées et les ontologies sont comme les sous-vêtements, tout le monde est d'accord sur le fait qu'ils sont nécessaires, mais personne ne veut utiliser le standard de quelqu'un d'autre. »³⁷. Derrière le caractère humoristique et anecdotique de la formule, le chercheur expose une réalité certaine pour les institutions patrimoniales : un musée, une bibliothèque, un centre d'archives et même les institutions de recherches, portent des regards très différents sur les documents. Les spécifications sont rattachées à des usages métiers ancrés dans le temps.

Durant le stage, ce fut une dimension à ne pas minimiser lors des réunions, engageant un dialogue constant entre les ingénieur(e)s et les archivistes du projet Lectaurep sur les atouts de la TEI pour Lectaurep. Les principaux reproches qui peuvent être fait à la TEI sont souvent basés sur des comparaisons avec la grammaire EAD et peuvent être en partie résumés ainsi :

37. S. V. Hooland, *L'application de l'intelligence artificielle au traitement de la bureautique et des mails*, programme du colloque, Colloque "Les Archives au défi du numérique" (17 et 18 octobre 2019), 2019, URL : <https://www.diplomatie.gouv.fr/fr/archives-diplomatiques/action-scientifique-et-culturelle/colloques-et-conferences/article/colloque-les-archives-au-defi-du-numerique-17-et-18-octobre-2019> on retrouve également une seconde version de cette anecdote dans F. Gillet, S. Hengchen, S. V. Hooland, *et al.*, *Introduction aux humanités numériques : méthodes et pratiques...*, pp.102.

- La philosophie de la TEI peut être éloignée des spécifications habituellement utilisées dans le monde documentaire tel que l'EAD. Dominique Stutzmann, chercheur du laboratoire IRHT³⁸ du CNRS, note à ce propos :

Les deux formats ont une philosophie différente. L'un est fait pour « enrichir » du « texte » (*Text Encoding Initiative*) et l'autre pour « décrire » des « archives » (je triche un peu, ici, puisque le mot *encoded* se trouve aussi dans *Encoded Archival Description*)³⁹.

Dès lors la mission de l'archiviste avec l'EAD est de « signaler » une ressource tandis que le chercheur, grâce à la TEI « étudie » cette ressource.

- La TEI peut également apparaître plus expressive que l'EAD et, de fait, moins enclue à représenter précisément les métadonnées essentielles qui entourent un document d'archives. L'EAD répond essentiellement à une norme archivistique et à un besoin « métier » : ISAD(G), et dispose d'un petit nombre d'éléments (seulement 150 contre 550 pour la TEI) qui permet une représentation idéale et un standard d'interopérabilité des fonds d'archives ;
- Cette norme est essentiellement conçue pour l'édition numérique de textes et répond à la complexité des normes de l'édition critique :

La complexité des normes de l'édition critique et la singularité de chaque projet éditorial semblent faire obstacle aux tentatives de standardisation que l'informatisation exige. La TEI, davantage adaptée aux sources littéraires que diplomatiques, permet de définir des schémas très différents et parfois difficilement interopérables pour des projets similaires.⁴⁰

Avant d'aborder les avantages qu'apporteraient la TEI à Lectaurep, nous pouvons rappeler que l'EAD, avant d'être un projet mis en application spécifiquement pour le domaine archivistique, a émergé du milieu universitaire (Université de Berkeley, 1993). De ce fait, de nombreux éléments de l'EAD sont empruntés à la TEI. On peut citer par exemple les éléments <eadheader> : <fildesc>, <titlestmt>, <publicationstmt>, <profiledesc>,

38. Institut de recherche et d'histoire des textes

39. D. Stutzmann, *EAD-TEI et TEI-EAD : quelques réflexions sur la conversion des notices de manuscrits médiévaux d'un format à l'autre*, Écriture médiévale & numérique, 2019, URL : <https://oriflamms.hypotheses.org/1715> (visité le 10/09/2020)

40. Camille Desenclos et Vincent Jolivet, « Diple, propositions pour la convergence de schémas XML/TEI dédiés à l'édition de sources diplomatiques », dans *Digital Diplomatics : The Computer as a Tool for the Diplomatist ?*, dir. Antonella Ambrosio, Sébastien Barret et Georg Vogeler, Cologne/Vienne/Weimar, 2014 (Archiv für Diplomatik, Schriftgeschichte, Siegel- und Wappenkunde. Beihefte, 14), pp. 23-30. in Olivier Canteaut, Olivier Guyotjeannin et Olivier Poncet, *Actes royaux et princiers à l'ère du numérique (Moyen Âge Temps modernes)*, PUPPA, 2020, URL : <https://acronavarre.hypotheses.org/2810> (visité le 07/09/2020), pp.61.

3.2 Le format pivot XML TEI, un choix réaliste ?

<creation>, ou <langusage>. De plus, le passage d'une spécification TEI à une autre peut être facilité par l'existence de transformations en XSLT^{41 42} basées sur des *cross-walks schema*⁴³. Dès lors, la TEI donnera naissance au format souhaité selon le résultat escompté.

Au-delà de la seule finalité éditoriale prêtée à cette grammaire, la TEI permet la préservation durable des données et des métadonnées. La TEI reste un format stable et rigoureux, ainsi pour être validé la syntaxe du langage XML doit être respectée, un schéma doit pouvoir être épinglé au document et vérifié par un éditeur XML comme *Oxygen*, et enfin le document TEI doit respecter le modèle des éléments présentés dans les *Guidelines TEI P5*.

41. XSLT ou *Extensible Stylesheet Language Transformations* est un langage de transformation basé sur le langage XML qui permet de transformer des fichiers XML ou HTML vers d'autres formats de fichier et vers d'autres spécifications.

42. Parmi ces transformations XSL on peut citer celle du projet HIMANIS pour convertir du XML EAD vers du XML TEI dans D. Stutzmann, *EAD-TEI et TEI-EAD : quelques réflexions sur la conversion des notices de manuscrits médiévaux d'un format à l'autre...* L'Université d'Oxford (*Bodleian Library*) pour le projet ENRICH (*European Networking Resources and Information concerning Cultural Heritage*) propose une transformation XML EAD vers XML TEI : University of Oxford - Bodleian Library, *ead2enrich*, URL : <http://projects.oucs.ox.ac.uk/ENRICH/XSLT/xsl/ead2enrich.xsl>. Le *Dutch Language Institute* propose une transformation XSL pour convertir du XML ALTO vers du XML TEI du : Dutch Language Institute (INL), *alto2tei*, URL : <https://github.com/INL/OpenConvert/blob/master/resources/xsl/alto2tei.xsl>

43. Un *crosswalk schema* est un mapping des champs d'une spécification vers une autre spécification, généralement présenté sous la forme d'un tableau d'équivalences. Par exemple, le passage du format MARC vers le *Dublin Core*, ou le passage de l'EAD vers la TEI. Pour voir d'autres utilisation du *crosswalk schema* : <http://www.ukoln.ac.uk/metadata/interoperability/>

3.2.3 ...qui a pourtant fait ses preuves dans des projets de standardisation : les avantages spécifiques pour Lectaurep

Certaines plate-formes, bénéficiant d'une bonne visibilité dans le monde documentaire et scientifique, font reposer leur chaîne éditoriale sur un format pivot XML-TEI. C'est le cas, par exemple, des plate-formes de dépôts d'articles scientifiques comme ISTE⁴⁴(projet de INIST/CNRS) et HAL⁴⁵(Hyper articles en ligne).

Dans le monde patrimonial, les plate-formes de transcription collaborative (*crowd-sourcing*), comme celle mise en place dans le cadre du projet Testaments de poilus⁴⁶ ou encore l'interface de recherche HIMANIS⁴⁷ structure également leurs informations recueillis en XML-TEI. Il y a donc une place pour la structuration des informations spécifiques à Lectaurep.

Afin de mettre en avant les avantages du fichier pivot XML-TEI appliqué aux ressources Lectaurep et comme format d'import-export privilégié accompagnant les images de répertoires dans eScriptorium nous citerons les points suivants :

1. La récupération des images dans *eScriptorium* peut être facilité par l'intégration des liens IIIF dans un format XML TEI qui peut s'effectuer avec les bons outils. En effet il existe de nombreux exemples d'articulation TEI-IIIF. En premier lieu, il est possible de recourir au service *Nakala* (Huma-num) qui implémente l'API image IIIF. On peut envisager de stocker les images du projet dans ce service et récupérer les manifestes IIIF en JSON correspondant à ces collections d'images de répertoires. Des *workshops*⁴⁸ organisée récemment autour du liage TEI-IIIF montrent qu'il est tout à fait possible d'implémenter des parties spécifiques du manifeste IIIF, ou encore le manifeste entier, dans un document XML-TEI. On peut relier une image à une page en intégrant l'URI IIIF correspondant dans un attribut @facs contenu dans un élément <pb> du fichier XML-TEI. On peut sinon relier la description d'un ensemble de documents d'archives à un manifeste IIIF qui contient une collection d'images, à l'aide d'un attribut @facs dans la partie <msDesc>. Enfin, il est possible d'utiliser une URI IIIF contenant une région spécifique de l'image correspondant

44. INIST/CNRS, *ISTEX et Conditor convertis au format TEI*, INIST, 2020, URL : <https://www.inist.fr/realisations/istex-et-conditor-convertis-au-format-tei/> (visité le 10/09/2020)

45. Laurent Capelli, Laurence Farhi et Laurent Romary, *A TEI conformant pivot format for the HAL back-office*, Text Encoding Initiative Conference and members meeting 2015, 2015, URL : <https://hal.archives-ouvertes.fr/hal-01221774> (visité le 10/09/2020)

46. <https://testaments-de-poilus.huma-num.fr/#/>

47. <https://www.himanis.org/>

48. Sur les différentes implémentations IIIF dans TEI voir Paolo Monella, *Linking Text and image : TEI XML and IIIF*, 2019, URL : <http://www1.unipa.it/paolo.monella/reires2019/> (visité le 10/09/2020) et la vidéo de Paolo Monella, *Linking text and image (TEI XML and IIIF)*, URL : <https://www.youtube.com/watch?v=Yu-eCBqVu9Y>, (consulté le 10/08/2020).

3.2 Le format pivot XML TEI, un choix réaliste ?

à une ligne de texte particulière dans un attribut `@facṣ` contenu dans un élément `<1b>`. Encore reste-t-il à trouver une chaîne de traitement adaptée au *back-office* de *eScriotorium* pour inclure les informations de ces manifestes ou de ces URI IIIF dans un grand nombre de fichiers XML-TEI produis lors de l'import et de l'export des images dans *eScriotorium*.

2. Si selon Michael Piotrowski⁴⁹, il est encore difficile de rattacher un document TEI à des outils de TAL, il souligne les efforts allant dans ce sens avec des outils comme *TXM* (un projet de textométrie de l'ENS Lyon), pour la fouille de texte ou *Textgrid* pour l'encodage des *tokens*. De plus rien n'empêche, un outil intégré, en Python par exemple, pour concevoir une *pipeline* sous la forme d'un script Python pour le traitement des fichiers XML TEI (avec des *packages* de traitement XML comme *lxml*⁵⁰ ou *Beautifulsoup*⁵¹) vers des tâches de TAL spécifiques (avec des *packages* de TAL comme *Spacy*⁵² ou *NLTK*⁵³ etc.).
3. Lectaurep peut concevoir par le biais d'un fichier XML-TEI des projets d'éditorialisation basés dans des interfaces *web* permettant de superposer le texte et l'image sur la base de fac-similés interactifs à l'image des « dossiers documentaires » proposés par la base *Theleme* de l'École nationale des chartes⁵⁴. De plus, on peut faire ressortir par un encodage en TEI la structure logique des tableaux d'un répertoire mais aussi des éléments plus précis comme la caractérisation de certaines mains d'écritures, typiques des graphies du XIX^e siècle, dans les répertoires de notaires. Ceci renforçant Lectaurep dans la possibilité d'offrir des contenus didactiques aux publics axés sur des dossiers d'études graphologiques ou des applications de « paléographie numérique ».
4. Anticiper les évolutions des standards archivistiques comme le modèle archivistique récent RiC (Records in Context), interopérable avec le *web* sémantique. Le fichier pivot TEI peut être vu comme une « passerelle » entre l'EAD et d'autres spécifications.
5. Pour les AN et le DMC, un fichier pivot XML-TEI est un moyen d'injecter, à terme, les métadonnées, rattachés aux traitements réalisés par les annotatrices et annotateurs dans *eScriotorium*, dans le système d'information de la SIV.

49. L. Romary, « Natural Language Processing for Historical Texts Michael Piotrowski (Leibniz Institute of European History) Morgan & Claypool (Synthesis Lectures on Human Language Technologies, edited by Graeme Hirst, volume 17), 2012, ix+157 p », *Computational Linguistics - MIT*, 40-1 (2014), p. 231-233, URL : <https://hal.inria.fr/hal-01016318> (visité le 10/09/2020)

50. *lxml 4.5*, URL : <https://lxml.de/>

51. *bs4 - Beautiful Soup 4.4.0*, URL : <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

52. *spaCy*, URL : <https://spacy.io/usage/spacy-101>

53. *nltk 3.5*, URL : <https://www.nltk.org/>

54. Theleme/École nationale des chartes, « Dossiers documentaires », URL : <http://theleme.ens.sorbonne.fr/dossiers/index.php>

CHAPITRE 3 : *Enjeux et analyse des problématiques liées aux données Lectaurep et au format pivot XML-TEI*

Le chapitre suivant décrit le *workflow* de travail mis en place pour le format pivot TEI. La figure 3.3, reprend les différentes étapes qui y seront présentées.

3.2 Le format pivot XML TEI, un choix réaliste ?

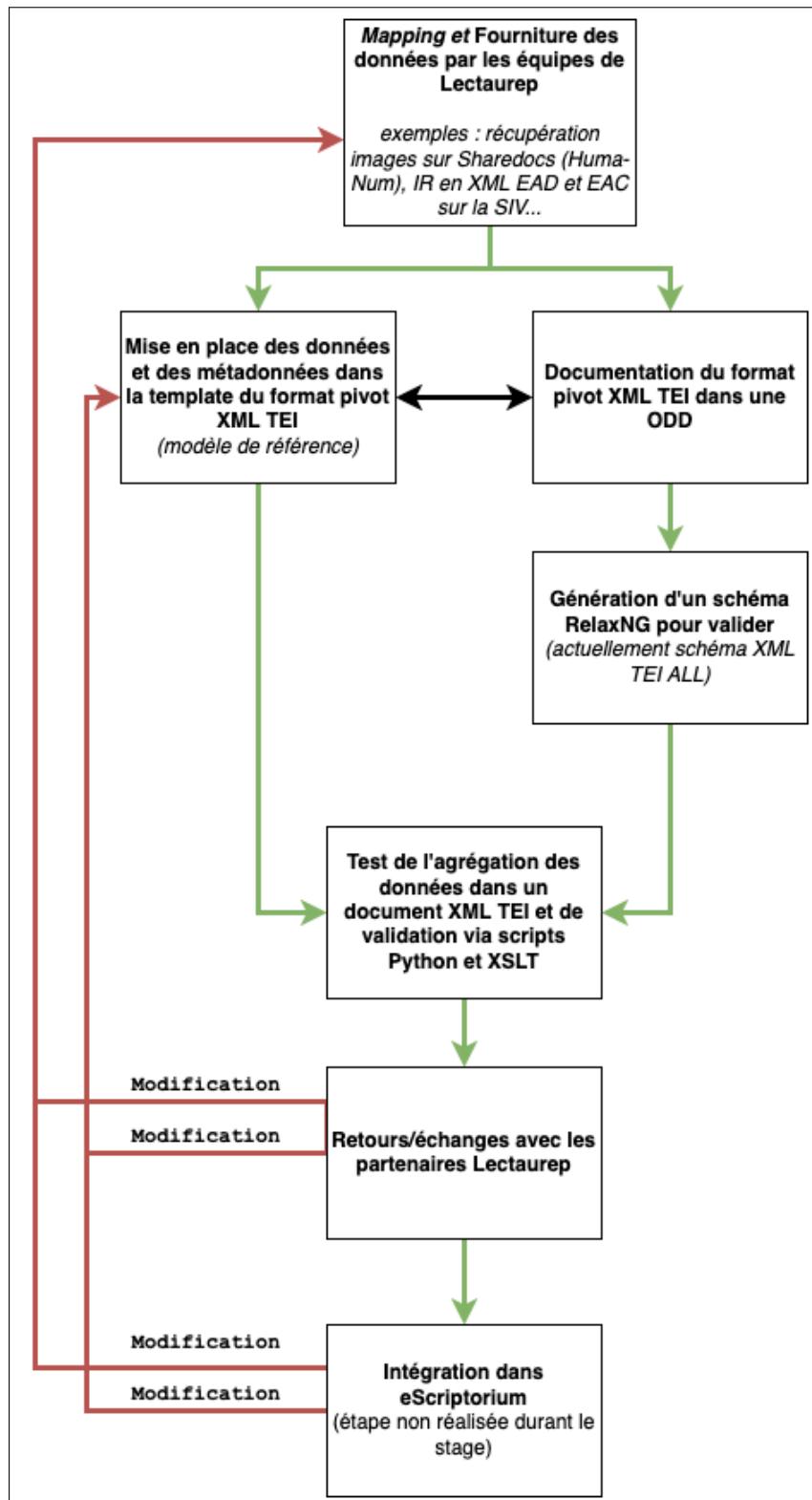


FIGURE 3.3 – Présentation du workflow pour le fichier pivot XML-TEI Lectaurep mis en place durant le stage ©L. Terriel, 2020, Diagrams.net

Chapitre 4

Workflow et mise en place d'une première version du fichier pivot XML-TEI

4.1 Un canevas de travail : un fichier XML-TEI dans un environnement partagé et ouvert

Une fois la modélisation des données élaborée et les objectifs du fichier pivot XML-TEI formalisés, il est nécessaire de concevoir le fichier pivot lui-même. Pour résumer et visualiser l'exposition des multiples données Lectaurep dans des champs TEI, il fallait disposer d'un fichier de travail XML suffisamment modulable et ouvert. Ceci pour synthétiser et inclure les nouvelles réflexions des acteurs qui évoluent au gré des discussions et des réunions concernant la récupération de données spécifiques à Lectaurep. Cela permet l'insertion ou la suppression des nouveaux éléments TEI.

On parle alors de mode de développement « agile »¹ pour la conception du pivot XML-TEI. Dès lors, nous avons décidé de travailler dans le cadre d'un canevas (*template*) XML-TEI, ou encore un fichier de travail XML-TEI prêt à l'emploi, pour autoriser un grand nombre de modifications à l'intérieur de ce fichier jusqu'à atteindre son état logique, stable et finalisé.

Pour créer la *template* XML-TEI, je suis parti d'un fichier XML-TEI (All) (Cf. Figure 4.1) généré à partir de l'éditeur XML *Oxygen*. Il s'agit d'une structure minimale (*framework*) en TEI comprenant un élément racine <TEI> ainsi que les deux sous-éléments

1. Il ne s'agit pas ici de mettre ici en place une méthode *Scrum*, répondant au manifeste agile avec la mise en place de tout un appareil de métiers comme un *scrum master*, un *product owner* etc. Je fais référence plutôt ici au moyen de communiquer et de faire évoluer le projet selon un système d'itérations périodiques pour rendre compte des évolutions d'un produit.

<teiHeader> et <text>, valide du point de vue d'un schéma Relax NG (REgular LAn-guage for XML Next Generation) englobant l'intégralité de la TEI.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml"
  schematypens="http://purl.oclc.org/dsdl/schematron"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Title</title>
      </titleStmt>
      <publicationStmt>
        <p>Publication Information</p>
      </publicationStmt>
      <sourceDesc>
        <p>Information about the source</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <p>Some text here.</p>
    </body>
  </text>
</TEI>
```

FIGURE 4.1 – Document XML-TEI minimal conforme et valide ©L. Terriel, 2020, *Oxygen XML Editor*

Pour mettre en relation et coordonner les membres du projet autour du canevas du format pivot XML-TEI et ainsi prévoir des retours de leurs part, il a été décidé de rendre accessible ce fichier (`template_pivot_TEI_lectaurep.xml`) via la plate-forme de gestion de versionnage GitLab² dans le dépôt rassemblant la documentation et les tests effectués à partir de ce format pivot XML TEI.

Cet environnement permet de rassembler les suggestions d'améliorations sous la forme de billets que l'on nomment *issues*. Cela permet également d'assurer une traçabilité des évolutions et des versions du format pivot, par le biais des révisions (*commits*) au fil des développements (soit un historique des modifications).

Durant le stage, nous avons eu finalement très peu de retours sur le format TEI. Cela peut s'expliquer par le contexte de télé-travail, l'attente de nouvelles fonctionnalités d'*eScriptorium* en priorité, les principes du langage XML pas entièrement maîtrisé et l'outil *GitLab* qui n'est pas encore bien partagé par tous les membres du projet Lectaurep. Le modèle étant encore abstrait à ce stade, certains acteurs ont du mal visualiser l'intégration du fichier XML-TEI dans *eScriptorium* ; les premiers tests d'export et d'import dans la plateforme feront sûrement évoluer ce contexte et la compréhension du fichier pivot.

2. La *template* du fichier pivot XML TEI pour Lectaurep est consultable dans les Annexes B, /B-Format_pivot_XML_TEI_Lectaurep/Doc/
Modélisation_et_documentation_format_pivot/template_pivot_TEI_lectaurep.xml

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

4.2.1 Repérage des données et règles préalables à l'encodage

Afin de proposer une première structuration de la *template* du fichier pivot XML-TEI, nous avons décidé, en coordination avec le DMC, de nous concentrer en premier lieu sur le repérage des données (*mapping*³) et la fourniture des données appartenant aux catégories suivante (Cf. Table 4.1 pour le détail des éléments). :

- **Métadonnées de gestion et de description essentielles** : correspondant aux données extraites des instruments de recherche en XML EAD et EAC fournis par le DMC et disponibles sur la SIV ;
- **Métadonnées techniques des images** : correspondant aux données contenues dans les images, répondant à la spécification EXIF ;
- **Images** : essentiellement les noms des fichiers images stockés sur un serveur Huma-num (ShareDocs) ;
- **Données provenant des traitements HTR** : correspondent aux transcriptions de vérité terrain (en l'absence de transcriptions HTR), exportées depuis sur la plate-forme eScriptorium, au format XML ALTO.

La première mise à plat des données proposée plus haut présente de multiples avantages dans la réalisation du format pivot.

On relève ainsi :

- la facilité pour accéder aux données dans le temps imparié ;
- des problèmes de données redondantes à rassembler ;
- une première hiérarchisation (Cf. Figure 4.2) des données dans le futur encodage XML-TEI, avec trois niveaux bien distincts, allant du plus au degré de granularité (métadonnées liées aux images et aux répertoires de notaires physiques) au plus bas, à savoir le texte de vérité terrain.

3. Nous entendons ici le *data mapping* comme le procédé qui consiste à extraire des données provenant de standards différents, qui seront exposées dans le format pivot XML TEI pour permettre leurs récupération dans d'autres systèmes d'informations.

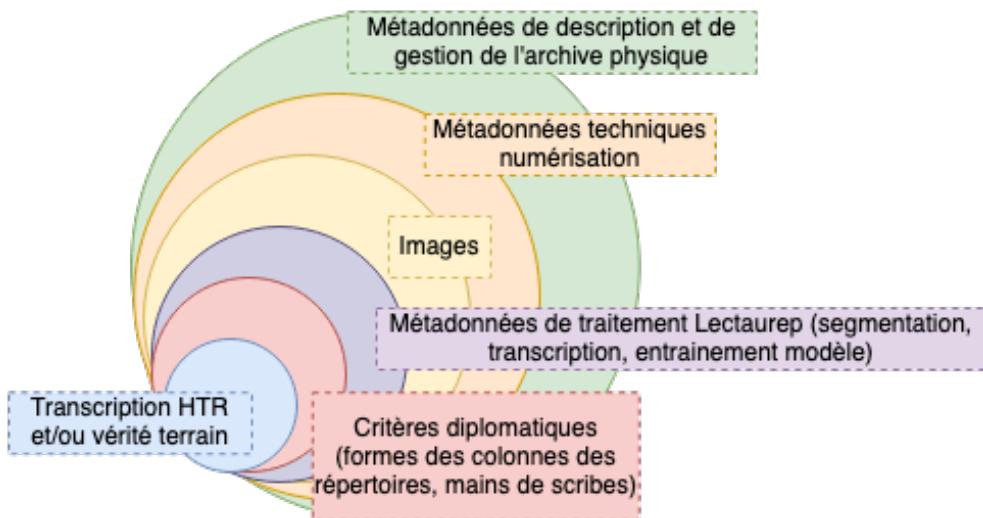


FIGURE 4.2 – Diagramme en oignon des différents niveaux de granularité des données Lectaurep pour représenter les imbrications dans le fichier pivot XML-TEI ©L. Terriel, 2020, Diagrams.net

TABLE 4.1 – *Mapping* des principales données Lectaurep utilisées pour la première version du fichier pivot XML-TEI

CATÉGORIES	SCHÉMAS	DONNÉES	ELEMENTS XML
Métadonnées de gestion et de description	EAD	Noms des fichiers EAD	<eadid>
		Nom et prénom du notaire	<persname>
		Étude du notaire	<corpname>
		Composant pour la description du niveau répertoire	<c level="series">
		Sous-composant pour la description du niveau minutes	<c level="recordgrp">
		Cotes des composants et des sous-composants de description	<unitid type="cote-de-consultation">
		Intitulé des composants et des sous-composants de description	<unittitle>
		Dates des composants et des sous-composants de description	<unitdate>
		Description physique des documents compris dans les composants et des sous-composants de description	<physdesc>

Continue sur la page suivante ↪

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

TABLE 4.1 – *Mapping* des principales données Lectaurep utilisées pour la première version du fichier pivot XML-TEI

CATÉGORIES	SCHÉMAS	DONNÉES	ELEMENTS XML
	EAC	Informations supplémentaires sur les composants et des sous-composants de description Dates d'exercices du notaire	<scopecontent> <dateRange> > <fromDate> / <toDate>
Métadonnées techniques des images numérisées	EXIF	Titre de la numérisation Version du standard Exif Nom du constructeur de l'équipement Nom du modèle de l'équipement Orientation de l'image en terme de colonnes et lignes	<EXIF:ImageDescription> <EXIF:ExifVersion> <EXIF:Make> <EXIF:Model> <EXIF:Orientation>
		Nombre de pixels par résolution (<EXIF:ResolutionUnit>) en fonction de la largeur	<EXIF:XResolution>
		Nombre de pixels par résolution (<EXIF:ResolutionUnit>) en fonction de la longueur	<EXIF:YResolution>
		Unité de mesure de <EXIF:XResolution> et <EXIF:YResolution>	<EXIF:ResolutionUnit>
		Date de dernière modification Position des composantes de chrominance par rapport à la composante de luminance. S'applique uniquement aux données compressées de type JPEG. La valeur par défaut est 1	<EXIF:ModifyDate> <EXIF:YCbCrPositioning>
		Date et heure du stockage de l'image comme donnée numérique	<EXIF:DateTimeDigitized>
		Date et heure du stockage de la création de l'image	<EXIF:DateTime>
		Informations spécifiques aux données compressées	<EXIF:ComponentsConfiguration>

Continue sur la page suivante ↪

CHAPITRE 4 : *Workflow et mise en place d'une première version du fichier pivot XML-TEI*

TABLE 4.1 – *Mapping* des principales données Lectaurep utilisées pour la première version du fichier pivot XML-TEI

CATÉGORIES	SCHÉMAS	DONNÉES	ELEMENTS XML
		<p>Si EXIF supporte Flashpix format Ver. 1.0, une valeur par défaut '0100' est inscrite sinon 'NULL'</p> <p>Spécifications sur l'espace colorimétrique de l'image</p>	<p><EXIF:FlashpixVersion></p> <p><EXIF:ColorSpace></p>
Données correspondant au document vérité terrain	ALTO	<p>Représentation de l'image</p> <p>Une page de répertoire</p> <p>Rectangle couvrant la zone imprimée d'une page</p> <p>Bloc de texte qui regroupe les lignes de textes</p> <p>Ligne de texte</p> <p>Forme de délimitation d'une ligne de texte, si elle n'est pas rectangulaire</p> <p>Contenu par <Shape>, décrit une forme polygonale</p> <p>Chaîne de caractères de la ligne de texte et leurs positions</p>	<p><Layout></p> <p><Page></p> <p><PrintSpace>; les coordonnées spatiales de l'élément et les dimensions est donné par les attributs @HPOS, @VPOS, @WIDTH, @HEIGHT</p> <p><TextBlock></p> <p><TextLine>, elle contient la <i>baseline</i> du texte sous la forme de points via l'attribut @BASELINE; les coordonnées spatiales de l'élément et les dimensions est donné par les attributs @HPOS, @VPOS, @WIDTH, @HEIGHT</p> <p><Shape></p> <p><Polygon>, les points sont décrits par l'attribut @POINTS</p> <p><String>, les chaînes de caractères de caractères sont comprises dans l'attribut @CONTENT; les coordonnées spatiales de l'élément et les dimensions est donné par les attributs @HPOS, @VPOS, @WIDTH, @HEIGHT identique à <TextLine></p>

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

L'étape qui suit le repérage des données consistait à répondre à la question : à quel élément TEI faire correspondre un élément *x* ou *y* de Lectaurep ?

Une solution que j'ai retenue consistait à commencer par le haut de l'arbre TEI puis à descendre petit à petit dans l'arborescence en prenant les éléments les uns après les autres. Cependant les éléments de Lectaurep appartenant à trois niveaux de granularités différents, nous devions disposer d'une architecture plus concrète que le canevas *TEI ALL* principal. Dès lors, j'ai pris le parti d'une architecture à trois niveaux (Cf. Figure 4.3) basés sur les trois catégories de données Lectaurep à encoder :

- la balise <teiHeader>⁴ pour encoder la partie la plus théorique, correspondant aux sources du répertoire c'est-à-dire les métadonnées relatives aux archives physiques et aux images (données issues des fichiers XML EAD/EAC et EXIF), les métadonnées liées à la production du document, les versions du fichier pivot et des indications sur la publication ;
- la balise <facsimile>⁵ pour recevoir les coordonnées (données issues des fichiers XML ALTO) et faire le pont entre les métadonnées générales du <teiHeader> et le <text>, par un système d'identifiants ;
- la balise <body>⁶ comprise dans la balise <text>⁷ pour accueillir le texte issu du document de vérité terrain, présent dans les fichiers XML ALTO.

```
<TEI>
  <teiHeader> // Décrit les documents et les images affiliées à la transaction
    <fileDesc>...</fileDesc> // Références aux documents d'archives structurés (EAD-EAC)
    <xenoData>...</xenoData> // Métadonnées techniques des images (Exif)
    <encodingDesc>...</encodingDesc> // Traces des spécificités et des évolutions du format pivot TEI
    <revisionDesc>...</revisionDesc> // Dates de révision de la template TEI
  </teiHeader>
  <facsimile>...</facsimile> // Représentation de la source sous forme de liens
  // vers les images et de coordonnées (ALTO)
  <text>
    <body>...</body> // Représentation de la source sous forme de texte
    // issu de la transcription vérité terrain ou issu de l'OCR;
    // et liens vers les images et les coordonnées (ALTO)
  </text>
</TEI>
```

FIGURE 4.3 – Schéma minimal retenu pour l'exposition des données Lectaurep dans un fichier pivot XML-TEI ©L. Terriel, 2020, *Oxygen XML Editor*

4. TEI, *TEI element teiHeader*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-teiHeader.html> (visité le 31/08/2020)

5. Id., *TEI element facsimile*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-facsimile.html> (visité le 31/08/2020)

6. Id., *TEI element body*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-body.html> (visité le 15/09/2020)

7. Id., *TEI element text*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-text.html> (visité le 31/08/2020)

4.2.2 Encodage des éléments de description des répertoires (EAD et EAC) dans le teiHeader

Une première approche du `teiHeader` consiste à modéliser grossièrement les flux entrants et sortants de métadonnées dans cette partie lors d'un import ou d'un export dans eScriptorium, quitte à affiner le travail par la suite (Cf. Figure 4.4). La plupart de ces données qui sont visibles dans les IR EAD et NP EAC-CPF trouveront naturellement leurs places dans le fichier pivot à partir de ces flux.

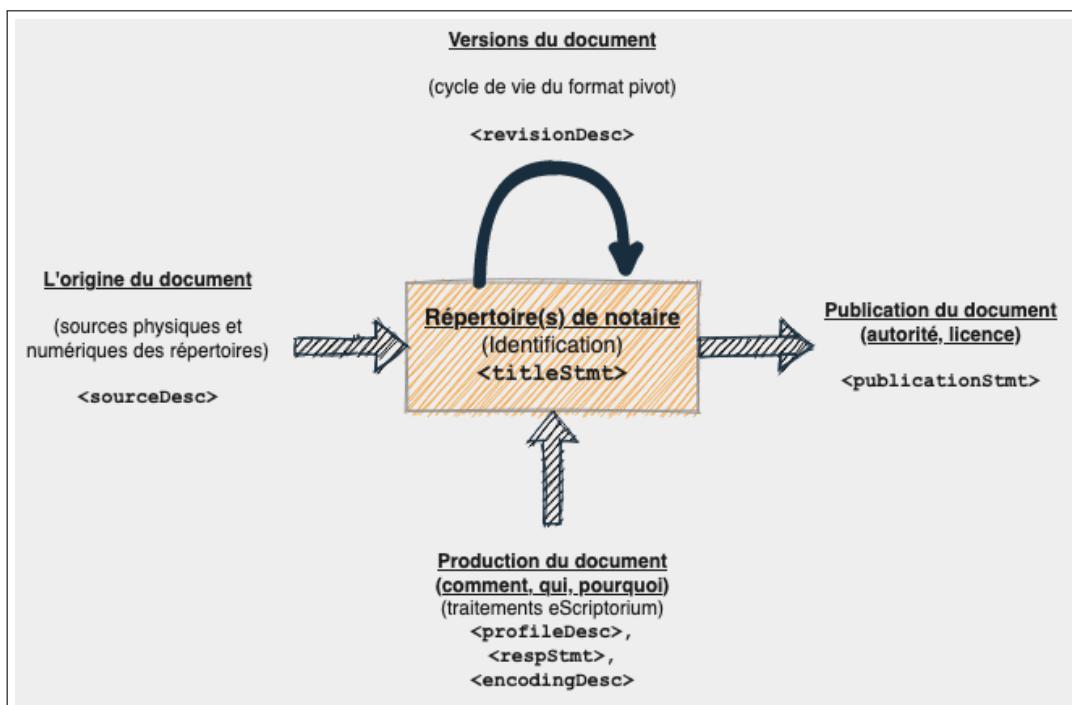


FIGURE 4.4 – Illustration des entrées et des sorties dans le `teiHeader` ©L. Terriel, 2020, diagrams.net (inspiré du document fourni par L. Romary, « The teiHeader at a glance »))

En comparant les deux IR « Images des répertoires du notaire M^eN » et « Minutes et répertoires du notaire M^eN » nous avons pu constater qu'un bon nombre d'informations se croisaient et qu'il était préférable de se baser sur l'IR « Images des répertoires du notaire M^eN », qui contient les informations spécifiques aux répertoires d'un notaire et les liens vers les images de ces derniers plutôt que celui contenant en sus des informations sur les minutes, ce qui aurait complexifié le travail par la suite.

Pour déterminer quels types d'éléments utiliser dans la conversion EAD vers TEI, je suis parti de l'existant et notamment des transformations EAD vers TEI réalisées dans le cadre d'« HIMANIS »⁸ et du projet « Enrich »⁹. Même si les ambitions d'encodage concernent des documents de natures différentes, certains éléments EAD de Lectaurep

8. D. Stutzmann, *EAD-TEI et TEI-EAD : quelques réflexions sur la conversion des notices de manuscrits médiévaux d'un format à l'autre...*

9. University of Oxford - Bodleian Library, *ead2enrich...*

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

peuvent être encodés de la même manière en TEI.

Dès lors certaines informations comme le nom et le prénom du notaire ont trouvés leurs places dans des éléments `<persName>`¹⁰ (déclinés en `<forename>` et `<surname>`), le numéro de l'étude à trouvé sa place dans l'élément `<title>`¹¹ (élément `<corpname>` en EAD) et les noms des fichiers d'IR EAD ont quant à eux été reportés dans le `<resp>`¹².

Un certain nombre de valeurs par défaut ont été ajoutées dans le `<publicationStmt>`¹³ comme l'organisation de conservation, l'adresse de l'institution etc. Le fait est que ces valeurs ne sont pas censées être modifiées au cours du projet.

La réelle difficulté s'est présentée lors de l'encodage des différents niveaux de description EAD des répertoires de notaires et du contenu de ces derniers (Cf. Figure 4.5). Un travail de dépouillement des arbres et de compréhension des logiques d'imbrications de ces IR, souvent complexes, a été indispensable. Généralement encodés dans le `<archDesc>`, il suivent la logique d'imbrications d'un niveau `<c>`, qui correspond à un répertoire de notaire, qui contient lui-même plusieurs `<c>` correspondant à des intervalles de pages du répertoire (par exemple l'intervalle des pages 124 r à 150 r¹⁴ contiennent une liste chronologique des actes pour la période du 2 janvier au 31 décembre 1877 au 31 décembre 1877).

10. TEI, *TEI element persName (personal name)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-persName.html> (visité le 16/09/2020)

11. Id., *TEI element title*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-title.html> (visité le 16/09/2020)

12. Id., *TEI element resp (responsibility)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-resp.html> (visité le 16/09/2020)

13. Id., *TEI element publicationStmt (publication statement)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-publicationStmt.html> (visité le 16/09/2020)

14. "r" signifiant "recto".

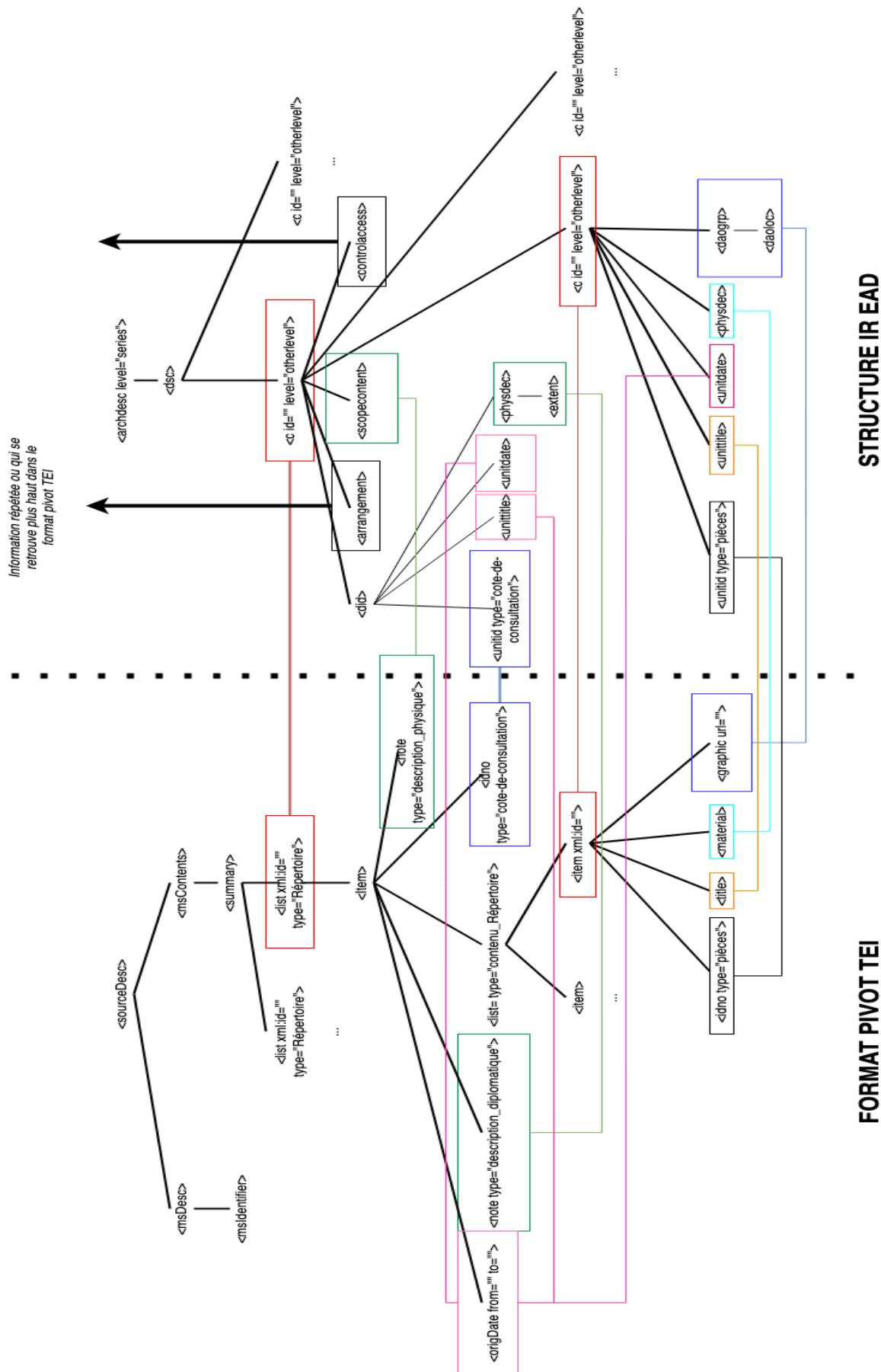


FIGURE 4.5 – Structure proposée en TEI pour l'encodage des différentes granularités descriptives des éléments EAD <c> pour représenter un répertoire et ses contenus.
 ©L.TERRIEL, 2020, Diagrams.net 90

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

Le modèle EAD suit un schéma tel que : le premier niveau de <c>, correspondant au répertoire lui-même, regroupe généralement la cote du répertoire (<unitid type="cote-de-consultation">), sa date (<unitdate>), sa description physique (<physdesc>), et des Informations supplémentaires sur les sources (<scopecontent>).

Les différents niveaux de <c> qui suivent, correspondent au contenu du répertoire. Ils regroupent alors le numéro des pages recto et verso (<unitid>), un titre qui est la liste des actes d'une période donnée (<unittitle>), le nombre de feuillets (<unittitle>), une description optionnelle (<scopecontent>) et une intervalle dans laquelle sont comprises les images numérisées se rapportant aux pages du répertoire (<daogrp> et <daoloc>).

À ce moment là, la difficulté fut de savoir si nous choisissons d'encoder un ou des répertoire(s) de notaire(s) lié(s) à une ou des transcription(s) vérité terrain dans le fichier pivot TEI ?

Je suis parti du principe que les utilisatrices ou utilisateurs peuvent effectuer des transcriptions de vérité terrain sur plusieurs répertoires d'un même notaire dans *eScriptorium*. Ainsi, j'ai décidé d'encoder tous les éléments descriptives consécutifs à l'ensemble des répertoires d'un notaire dans le fichier pivot XML-TEI.

La structure TEI retenue pour l'encodage des descriptions de répertoire est reportée dans la Figure 4.5. Dans un <SourceDesc>¹⁵ et un <msDesc>¹⁶ (permettant de décrire la source de manuscrits), on dispose d'un <msIdentifier>¹⁷ commun à l'ensemble des répertoires d'un même notaire. Habituellement, réservé à la description d'un manuscrit, le <msIdentifier> fait ici office de carte descriptive unique pour plusieurs répertoires étant donné que les informations sont des valeurs par défaut (pays, lieu, institution, lieu de dépôt) similaires pour les répertoires. On évite donc le phénomène de redondance d'informations. Suit alors le <msContents>¹⁸ qui rassemble le premier niveau de <c> (EAD) correspondant à un répertoire et les différents niveau de <c> (EAD) imbriqués qui suivent.

Le premier niveau est alors encodé dans une première liste <list>¹⁹ portant un attribut @type « répertoire » et les niveaux inférieurs de <c> sont encodés dans une seconde liste <list> caractérisée par un attribut @type « contenu_Répertoire » qui comporte des

15. Id., *TEI element sourceDesc (source description)*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-sourceDesc.html> (visité le 16/09/2020)

16. Id., *TEI element msDesc (manuscript description)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-msDesc.html> (visité le 16/09/2020)

17. Id., *TEI element msIdentifier (manuscript identifier)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-msIdentifier.html> (visité le 16/09/2020)

18. Id., *TEI element msContents (manuscript contents)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-msContents.html> (visité le 16/09/2020)

19. Id., *TEI element list*, URL : <https://tei-c.org/Vault/P5/current/doc/tei-p5-doc/fr/html/ref-list.html> (visité le 16/09/2020)

<item>²⁰ caractérisant les différentes séries de pages pour des périodes d'actes définies. La structure TEI comprise dans les <item> suit ensuite l'ordonnancement logique des descriptions de l'EAD.

Il y a un décalage à noter : le premier élément <list> (TEI) correspond au premier <c> (EAD) et les éléments <item> de la seconde <list> (TEI) correspondent aux différents <c> (EAD) inférieurs. Dans le format pivot XML-TEI cela se traduit par une premier élément <list> (TEI) portant les métadonnées du répertoire, comme une sorte de « meta-liste » (Cf. Figure 4.6), et un second élément <list> (TEI) ne faisant que rassembler items compris comme les série de pages de répertoires.

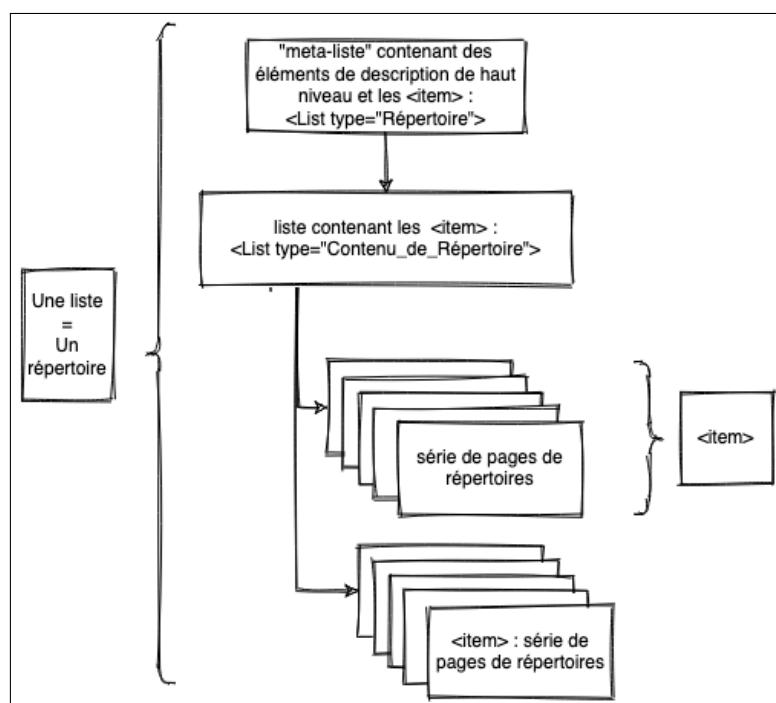


FIGURE 4.6 – Imbrication des deux niveaux de <list> dans le <teiHeader> ©L. Terriel, 2020, Diagrams.net

Quelques points restent à détailler. La balise EAD <arrangement> n'a pas été prise en compte car elle n'apparaissait pas immédiatement comme un élément indispensable d'encodage pour le projet. Quant à la balise <contolaccess>, répétée à chaque premier niveau de <c>, elle correspond au genre administratif relié à un référentiel des AN. Il nous a paru préférable de rapporter cette information en une seule fois dans une balise TEI <additional>²¹ contenant une balise <bibl>²² avec un attribut @source, sur lequel on

20. Id., *TEI element item*, URL : <https://tei-c.org/Vault/P5/current/doc/tei-p5-doc/fr/html/ref-item.html> (visité le 16/09/2020)

21. Id., *TEI element additional*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-additional.html> (visité le 16/09/2020)

22. Id., *TEI element bibl (bibliographic citation)*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-bibl.html> (visité le 16/09/2020)

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

rapporte le type de référentiel associé. Elle contient elle-même deux balises `<title>` avec un attribut `@xml:id`, reliant l'entité du référentiel au format pivot. La balise reçoit alors le titre administratif tel qu'il est décrit dans le référentiel des AN, à savoir « répertoire d'officier public ministériel ».

Concernant les notices producteurs en EAC-CPF contenant des informations relatives aux notaires (personnes) d'une même étude dans le pivot XML-TEI, la stratégie envisagée fût la suivante : dans le `<msItemStruct>`²³ nous avons choisi de rapporter un minimum d'informations relatives aux notices producteurs en pointant à l'aide d'attributs `@source`, placés sur un `<persName>` pour le notaire et sur un `<orgName>` pour l'étude. Cet attribut pointe alors directement vers les identifiants des notices producteurs, qui pointent elles mêmes vers d'autres référentiels ainsi que des vocabulaires contrôlés internes aux AN.

La conversion de la TEI vers l'EAD a pu poser certaines difficultés majeures. Les différents étages d'imbrication présents dans les IR en EAD des répertoires n'a pas facilité la tâche, et la redondance de certaines informations a obligé un travail souvent fastidieux d'aller-retour entre les deux IR EAD. Il y a fort à parier que la structure envisagée, du fait de sa complexité et de son niveau d'abstraction actuel, évoluera grandement dans la suite du projet. Cependant, cette première structure pose les bases d'une première correspondance entre des éléments des IR et des NP vers la TEI, que les utilisateurs pourraient souhaiter retrouver dans leurs imports et exports d'images dans eScriptorium.

4.2.3 Encodage des éléments des métadonnées EXIF dans le teiHeader via des éléments xenoData

Dans la TEI il n'existe pas d'élément pour décrire la sémantique propre aux métadonnées techniques de numérisation EXIF (Cf. Figure 3.2). De plus, il n'est pas conforme aux recommandations de la TEI, d'utiliser un élément existant est de lui spécifier une fonction autre que celle d'origine. Pour donner un exemple d'usages non conforme à la TEI dans ce contexte : `<item type="EXIF">` ou encore `<title type="EXIF">`.

Comme Lou Burnard le souligne, c'est une manière de « dupliquer » la fonction d'éléments TEI existants²⁴. Pour certains schémas comme le *Dublin Core*, la TEI a anticipé dans son en-tête un certain nombre d'équivalences comme avec le `<DC:title>` et le `<title>` TEI. Mais on pourrait envisager dans un projet d'encodage, et cela pour de nombreuses raisons, que l'élément `<title>` TEI ne correspond pas assez bien à la sémantique de l'élément *Dublin Core*. On considère alors que l'élément `<DC:title>` est un

23. Id., *TEI element msItemStruct (structured manuscript item)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-msItemStruct.html> (visité le 16/09/2020)

24. L. Burnard, « What is TEI Conformance, and Why Should You Care ? », *Journal of the Text Encoding Initiative*, 2019-2020-12 (2020), URL : <http://journals.openedition.org/jtei/1777> (visité le 16/09/2020)

élément non-TEI et on souhaiterait une extension pour ce dernier élément, à inclure dans un schéma TEI.

Afin d'éviter ces problèmes d'intégrité et de garantir une liberté dans l'encodage, la TEI a prévu l'accueil d'autres vocabulaires XML par un élément <xenoData>²⁵, qui permet d'introduire une extension pour des métadonnées provenant d'un autre schéma.

Son fonctionnement est le suivant (Cf. Figure 4.7) : pour une image, un élément TEI <xenoData> porte dans un attribut @xmlns:exif l'espace de nom (*namespace*) Exif²⁶. Dès lors, aux niveaux inférieurs, on peut récupérer les différents champs du schéma Exif, en créant des balises précédées d'un préfixe exif:. Un élément <exif:Exif> est requis entre <xenoData> et les métadonnées qui suivent pour permettre la validation du document TEI final.

```

1  <xenoData facs="#FRAN_0025_0046_L-0" n="FRAN_0025_0046_L-0.jpg"
2    xmlns:exif="http://ns.adobe.com/exif/1.0/">
3    <exif:Exif>
4      <exif:ColorSpace>
5        65535
6      </exif:ColorSpace>
7      <exif:DateTimeDigitized>
8        2015:09:07 09:33:02
9      </exif:DateTimeDigitized>
10     <exif:ImageDescription>
11       47 - Main frame
12     </exif:ImageDescription>
13     <exif:Make>
14       i2S, Corp.
15     </exif:Make>
16     <exif:Model>
17       SupraScanII [SN: 283910] - Cam7600RGB [SN: 283910]
18     </exif:Model>
19   </exif:Exif>
</xenoData>
```

FIGURE 4.7 – Un exemple d'encodage d'informations EXIF dans une balise TEI <xenoData> ©L. Terriel, 2020, Oxygen XML Editor

25. TEI, *TEI element xenoData (non-TEI metadata)*, URL : <https://tei-c.org/release//doc/tei-p5-doc/en/html/ref-xenoData.html> (visité le 16/09/2020)

26. Nous avons choisi l'espace de nom EXIF de l'entreprise Adobe, URL : <https://github.com/adobe/xmp-docs/blob/master/XMPNamespaces/exif.md> mais il est tout à fait possible d'en utiliser d'autres comme <http://cipa.jp/exif/1.0/> ou encore <http://www.w3.org/2003/12/exif/ns>; Cependant, il faut faire attention à l'usage des préfixes qui peuvent varier.

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

4.2.4 Encodage des éléments ALTO correspondant aux zones et lignes de texte, et à la transcription dans l'élément `facsimile` et l'élément `body`

Il n'a pas été clairement défini durant le stage si le fichier pivot XML-TEI devait pointer vers un fichier XML ALTO seul (*standalone*)²⁷. Dans ce contexte, j'ai opter pour l'encodage des éléments ALTO dans le fichier pivot TEI.

L'encodage des fichiers XML ALTO qui contiennent à la fois les coordonnées de polygones (élément ALTO `<Polygon>`) entourant les zones de texte et de la ligne de base (@BASELINE de l'élément ALTO `<TextLine>`), ainsi que le texte (attribut @CONTENT de l'élément ALTO `<String>`) répond à la problématique suivante : comment envisager une structure qui permet de relier le texte à sa représentation spatiale et à l'image ?

La mise à plat de l'arbre ALTO a permis de déterminer deux niveaux TEI bien distincts (Cf. Figure 4.8) :

- une structure `<facsimile>` qui contient les représentations des répertoires des notaires sous la forme d'images. Chaque image est encodée dans un élément TEI `<surface>`²⁸ qui constitue une surface en deux dimensions, correspondant à l'image de la page transcrise contenue dans un élément `<graphic>`²⁹.

Des chercheurs du département de langue allemande et de littérature de l'Université d'Innsbruck en Autriche (qui ont travaillé dans le cadre du projet READ/Transkribus) rappellent que la connexion de l'image de la page avec la transcription n'est pas forcément évidente en TEI. Les méthodes diverges dans la pratique³⁰, cependant l'élément `<facsimile>` permet de représenter via des `<zone>`³¹ les différents niveaux de représentation ALTO d'un document véritable terrain : `<PrintSpace>` pour le rectangle couvrant la zone imprimée d'une page, `<TextBlock>` pour le bloc paragraphe de texte, `<TextLine>` pour une ligne de texte, `<String>` pour contenir les mots et `<Shape>/<Polygon>` pour la forme du texte, sont reportés dans des attributs.

27. La connexion du fichier TEI et du fichier ALTO pouvant se faire par l'intermédiaire d'un attribut qui pointe vers le fichier ALTO dans le TEI ou d'un fichier METS (*Metadata Encoding and Transmission Standard*). A. Belaïd, I. Falk et Y. Rangoni, « Représentation des données en XML pour l'analyse d'images de documents »...

28. TEI, *TEI element surface*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-surface.html> (visité le 16/09/2020)

29. Id., *TEI element graphic*, URL : <https://tei-c.org/release/doc/tei-p5-doc/en/html/ref-graphic.html> (visité le 16/09/2020)

30. Günter Mühlberger, Philip Kahle et Sebastian Colutto, « Preprint : Handwritten Text Recognition (HTR) of Historical Documents as a Shared Task for Archivists, Computer Scientists and Humanities Scholars. The Model of a Transcription Recognition Platform (TRP) » (, 2014), URL : https://www.academia.edu/8601748/Preprint_Handwritten_Text_Recognition_HTR_of_Historical_Documents_as_a_Shared_Task_for_Archivists_Computer_Scientists_and_Humanities_Scholars_The_Model_of_a_Transcription_and_Recognition_Platform_TRP_ (visité le 14/09/2020), pp.3

31. TEI, *TEI element zone*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-zone.html> (visité le 16/09/2020)

buts @type sur les éléments TEI <zone>.

A noter que les *Guidelines TEI* recommandent l'utilisation d'un élément <zone> plutôt que d'un élément <path>³² pour spécifier des lignes comprenant plus de deux points de coordonnées, comme dans le cas des polygones. Les coordonnées spatiales des différents niveaux de représentation de la page sont encodés dans des attributs @ulx, @uly, @lrx et @lry sauf pour l'élément <zone> décrivant le polygone qui utilise un attribut @points spécifique et qui comprend plusieurs valeurs numériques par pairs (*x,y*) décrivant les points par lesquels passe la ligne entourant les mots ou les phrases.

- une structure plus simple comprise dans l'élément TEI <body> de l'élément <text>, permet de récupérer la transcription ALTO. Cette structure s'appuie sur celle proposée par le projet du Dutch Language Institute (INL)³³. La logique est la même que pour le <facsimile> et ces différents niveaux d'imbrications. Cependant la sémantique des balises TEI se concentre sur la représentation de la vérité terrain. Ainsi l'élément <div>³⁴ fait référence à la <surface> et contient systématiquement : un élément <ab>³⁵ qui correspond à un bloc de texte (TextBlock (ALTO)) contenant un élément <w>³⁶ dans lequel on ajoute les données textuelles et un élément <lb>³⁷ qui marque le début d'une nouvelle ligne et qui correspond à la (TextLine (ALTO)).

32. Id., *TEI element path*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-path.html> (visité le 16/09/2020)

33. Dutch Language Institute (INL), *alto2tei...*

34. TEI, *TEI element div (text division)*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-div.html> (visité le 16/09/2020)

35. Id., *TEI element ab (anonymous block)*, URL : <https://tei-c.org/release/doc/tei-p5-doc/en/html/ref-ab.html> (visité le 16/09/2020)

36. Id., *TEI element w (word)*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-w.html> (visité le 16/09/2020)

37. Id., *TEI element lb (line beginning)*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-lb.html> (visité le 16/09/2020)

4.2 Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage

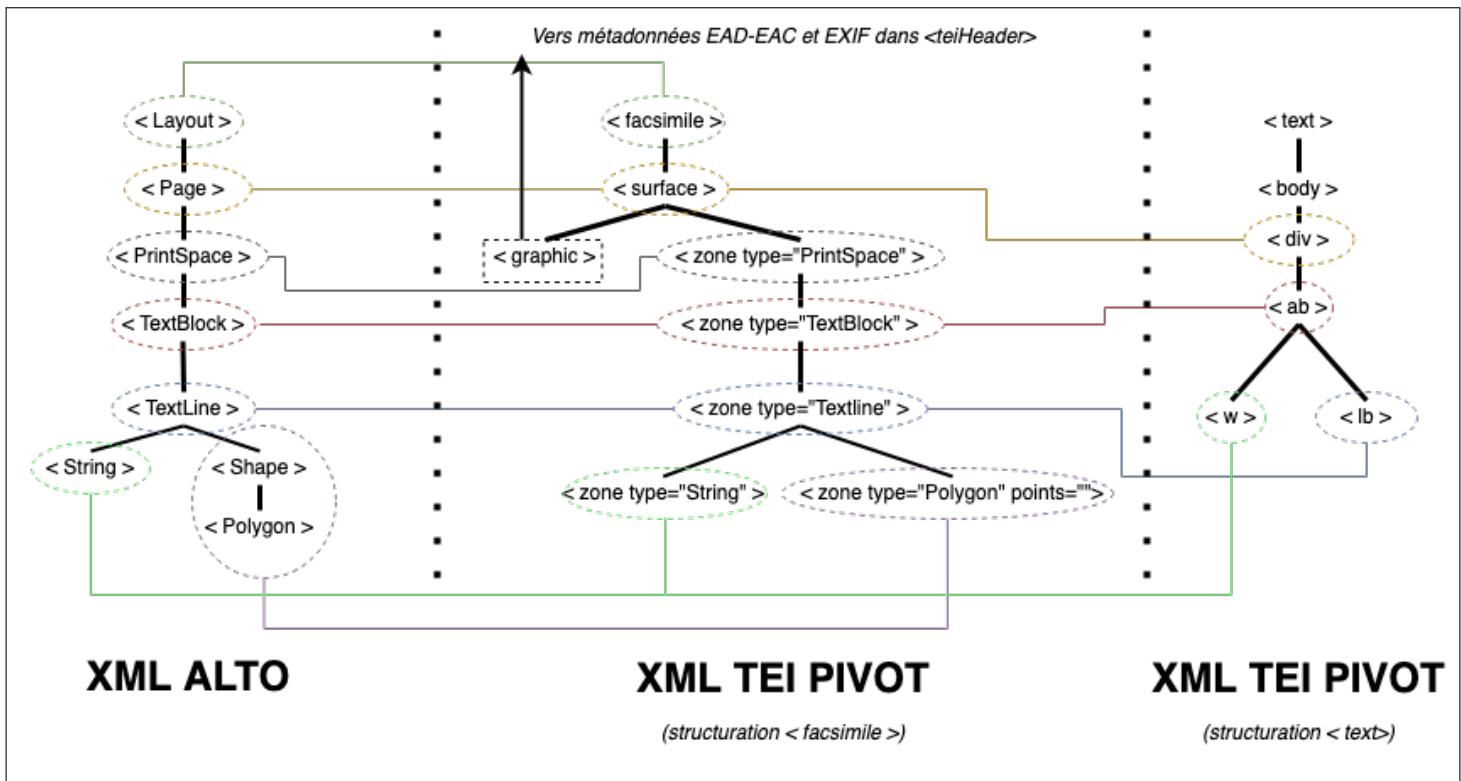


FIGURE 4.8 – Structure du fichier XML ALTO à gauche et du fichier XML TEI pivot et liens entre les éléments ©L. Terriel, 2020, Diagrams.net

Les deux niveaux pour encoder la segmentation et le texte transcrit sont finalement reliés entre eux, mais également à la partie du `<teiHeader>`, par un système d'identifiants de type `@url`, `@facs` et `@xml:id`.

Cette structure à deux niveaux doit encore évoluer dans la suite du projet. En effet les images comprises dans des éléments `<graphic>` du `<facsimile>` devront être adaptées pour recevoir les liens du manifeste IIIF (actuellement ce sont les images stockées en local). De plus, la structure du `<text>` devra encore être modifiée pour accueillir les futures transcriptions HTR (actuellement je n'ai pu traiter que les vérités terrains), l'étiquetage des entités nommées relevées dans les transcriptions HTR et la structure logique des tableaux des répertoires.

4.2.5 Compléments sur l'encodage du fichier pivot XML-TEI

Dans un souci de traçage des modifications ou des révisions et pour spécifier des points particuliers d'encodage dans le fichier pivot XML-TEI, il nous a paru judicieux de disposer d'un `<encodingDesc>` suivi d'un `<editorialDecl>` et d'un `<revisionDesc>` pour y inclure ces informations.

4.3 Une ODD pour documenter, partager et valider le canevas XML-TEI

Un format pivot encodé en TEI pour rentrer dans le cadre de l'échange de fichier et être traité par différents systèmes d'informations, doit répondre aux principes de la *TEI-conformance*³⁸, à savoir :

- Le document doit respecter les principes du langage XML bien formé ;
- L'encodage doit pouvoir être validé du point de vue d'un schéma *TEI-ALL* ou d'un schéma personnalisé qui répond au modèle abstrait de la TEI ;
- Enfin l'encodage doit être documenté.

En effet, le schéma que nous proposons pour Lectaurep est en somme une personnalisation de la TEI, pour laquelle il faut rendre compte d'une documentation.

Afin de générer cette documentation nous nous sommes appuyé sur le format de spécification TEI ODD³⁹ (*One Document Does it all*) qui permet en outre de :

- Générer un schéma RelaxNG ainsi qu'une documentation associée ;
- Définir les éléments utilisés dans le cadre du format pivot ;
- Typer les éléments, définir les séquences d'enchaînement des éléments, choisir le type d'attribut à fixer sur les éléments ;
- S'appuyer sur les macros et les modules de la TEI P5, pour restreindre le nombre d'éléments à utiliser.

Pour générer l'ODD spécifique au projet j'ai utilisé une feuille de style `oddbyexample.xsl` fournie par la TEI, pour générer la documentation à partir du canevas du fichier pivot XML TEI, à savoir la source `template_pivot_TEI_lectaurep.xml`, dans les formats suivants : XML (format natif), HTML, pour une visualisation *web*, et PDF⁴⁰.

38. Jean-Baptiste Camps, *Structuration des données et des documents : balisage XML - Personnaliser la TEI : One Document Does it all*, cours, 2017, URL : https://halshs.archives-ouvertes.fr/cel-01706530/file/06_TEI_ODD_Camps_20170202.pdf

39. TEI, *TEI : Getting Started with P5 ODDs*, URL : <https://tei-c.org/guidelines/customization/getting-started-with-p5-odds/>

40. L'ODD du projet (dans les formats cités) est disponible dans les Annexes, `/B-Format_pivot_XML_TEI_Lectaurep/Doc/Modélisation_et_documentation_format_pivot/`

4.4 Les évolutions du fichier pivot XML-TEI pour la suite du projet

Le fichier pivot XML-TEI étant en cours de développement, et devant amener encore son lot de modifications, il nous a paru préférable de ne pas rentrer dans le détail des modules TEI. Ceci devant constituer une prochaine étape vers la constitution d'un schéma RelaxNG propre au projet, celui-ci étant encore le schéma *TEI-ALL*.

Cette « ODD de travail » rappelle les objectifs actuels, rend visible le canevas de la présente version du format pivot (pour les personnes ne disposant pas d'un éditeur XML, par exemple), de suggérer des axes de travail pour la suite et enfin de rendre accessible la documentation sur les éléments utilisés. L'ODD fournie au format HTML, pourrait tout à fait être partagée sur le blog *hypothèses* Lectaurep dans le cadre d'une diffusion des savoirs-faire et dans la logique de retours d'expériences.

4.4 Les évolutions du fichier pivot XML-TEI pour la suite du projet

Pour la suite à donner au fichier pivot XML-TEI, les cas suivants ont pu être envisagés :

- Trouver le moyen d'intégrer les liens vers les manifestes IIIF et les URI vers les images dans les éléments TEI `<graphic>`, et tester leurs récupération ;
- Envisager une structure d'accueil TEI pour les futures transcriptions HTR ;
- Proposer un balisage pour les entités nommées dans la future transcription HTR en se basant sur les éléments TEI au niveau de la balise `<body>` afin d'étiqueter les personnes (`<person>`), les prix (`<num type="prix_acte">`), les lieux (`<placeName>`), les dates (`<date>`), les professions (`<roleName type="metier">`), les titres honorifiques (`<roleName type="honorifique">`) etc. Là encore tout dépend du niveau d'information que l'on souhaite récupérer et des choix de balises qui seront faits⁴¹ ;
- Le fichier pivot XML-TEI devra faire ressortir, à terme les logiques de structures internes des colonnes des tableaux des répertoires, notamment pour développer des éditions « paléographiques ». Sur ce dernier point, Laurent Romary, directeur de recherche pour INRIA et qui a suivi la constitution du fichier pivot XML-TEI, a suggéré, dans un premier temps, de rendre compte de la sémantique des colonnes des tableaux des répertoires (ce sur quoi travaille actuellement *eScriptorium*).

41. Solenn Le Pevedic et Denis Maurel, « Retour sur les annotations des entités nommées dans les campagnes d'évaluation françaises et comparaison avec la TEI », *Corela. Cognition, représentation, langage*, 14–2 (2016), URL : <http://journals.openedition.org/corela/4644> (visité le 16/09/2020)

Par exemple, une colonne correspond au type d'acte, la suivante au prix, la suivante au client etc. ; dans un deuxième temps, il faudrait réunir les blocs horizontaux similaires si ces blocs concernent un même acte. Le dernier point constitue une tâche difficile dans la mesure où les répertoires de notaires présentent des écritures pouvant parfois déborder des colonnes ou prendre de la place sur plusieurs lignes. Toujours est-il que des outils existent comme *Grobid*⁴² (*GeneRation Of BIbliographic Data*) pour extraire des informations sur la structure d'un document, parser et restructurer ces informations dans un fichier XML-TEI ; un tel outil nécessiterait cependant des manipulations avant d'être exploitable sur les répertoires ;

- Approfondir les logiques d'écritures propres au XIX^e siècle dans les répertoires : il serait envisageable de relier certains éléments de mise en forme du texte dans le répertoire à un référentiel de mains génériques spécifique et caractéristique des répertoires de notaires dans le <**msDesc**> contenant des éléments <**handDesc**> ;
- Dans une logique d'entonnoir, au fur et à mesure de l'avancement des choix de balisages, il faudra formaliser ces règles dans l'ODD afin de restreindre les modules TEI pour générer un schéma RelaxNG propre au projet ;
- Effectuer des tests d'intégration du format pivot XML TEI dans la plate-forme *eScriptorium* pour permettre des retours utilisateurs et des tests fonctionnels de compatibilité.

42. Grobid documentation, URL : <https://grobid.readthedocs.io/en/latest/Introduction/>

4.5 Simuler l'agrégation des données et la validation d'un fichier XML-TEI pivot avec un *script* Python

4.5 Simuler l'agrégation des données et la validation d'un fichier XML-TEI pivot avec un *script* Python

Certains acteurs du projet Lectaurep ressentaient le besoin de visualiser les données provenant des sources XML EAD-EAC, ALTO et des images dans la première version du canevas XML-TEI.

Cependant, un encodage « à la main » des données l'une après l'autre dans un fichier XML-TEI était impensable pour des données aussi nombreuses. Pour réaliser cette tâche, j'ai imaginé un *script* pour permettre le traitement automatique de quelques fichiers Lectaurep afin d'extraire les informations des fichiers, les structurer dans l'arborescence du pivot TEI souhaitée et valider cette arborescence par le biais d'un schéma de validation (*TEI-ALL*).

J'ai donc réalisé un outil, sous la forme d'un CLI (*Command Line Interface*) appelé *Generator Lectaurep-TEI*, basé sur des *scripts* en langage Python pour mener à bien cette tâche. Parmi eux, un *script* d'exécution principal `main.py` chargé de dérouler les différentes étapes du programme, une série de trois modules asservis au script principal : `build_utils.py`, `extract_utils.py`, et `validation_utils.py` contenant des fonctions utiles pour réaliser des tâches précises durant le processus de traitement des fichiers ainsi que deux feuilles de transformation rédigées en XSLT : `Lectaurep_ALTO2TEI.xsl` et `Lectaurep_EADEAC2TEI.xsl`.

L'objectif étant de parser et de restructurer les fichiers XML en entrée pour récupérer les informations dans une arborescence TEI en sortie. Nous reviendrons sur ces feuilles de styles dans la section suivante.⁴³.

Afin de tester le programme nous avons formé un jeu de données autour des fichiers de répertoires du notaire Legay (`Set_test_Legay`). Ce set de données est constitué d'un dossier `Data_xml_alto` qui contient les fichiers XML ALTO, d'un dossier `Data_xml_ead_eac` contenant les fichiers XML EAD-EAC et enfin d'un dossier `images` rassemblant les images de répertoires. Il s'agit là de la configuration de fichiers dont les utilisatrices et les utilisateurs doivent disposer pour utiliser l'outil.

43. Le programme est composé des *scripts* Python, des feuilles de transformation XSL, des fichiers de tests, des schémas de validation, et d'une documentation d'installation et de fonctionnement « lisez-moi » (`readme.md`) est disponible dans l'Annexe B,
`/B-Format_pivot_XML_TEI_Lectaurep/Generator_Lectaurep2TEI`

Pour fonctionner, les *scripts* nécessitent un environnement virtuel basé sur Python 3 et dans lequel seront installés :

- *lxml* (version 4.5) : il permet de manipuler des fichiers XML et HTML. Il est utilisé ici pour lire des schémas de validation (DTD, RelaxNG, XMLSchema, Schematron) ;
- *beautifulsoup4* (version 4.4) : autre parseur de fichiers HTML et XML. Il est utilisé pour sa fonction de manipulation des arbres XML ;
- *pyfiglet* (version 0.8.0), *tqdm* (version 4.46.1) et *termcolor* (version 1.1.0) : une série de *packages* utilisés pour obtenir un visuel spécifique (couleur des messages, barres d'avancement etc.) sur la progression des tâches.

Les autres *packages* sont déjà inclus dans la distribution Python (modules et fonctions *built-in*). Tous ces *packages* Python sont listés dans le fichier `requirements.txt` pour faciliter l'installation. Une documentation qui résume les étapes d'installation et d'utilisation est disponible dans un fichier de type « lisez-moi » (`readme.md`).

4.5.1 Les différentes étapes du programme : le script principal `main.py`

Avant de concevoir le programme j'ai modélisé un algorithme afin de diviser les tâches de travail et pour résumer chaque étape du programme permettant d'accélérer considérablement le développement de l'outil (Cf. Figure D.2 en Annexes).

Le programme réalise les étapes suivantes :

1. L'usager entre une ligne de commande pour indiquer l'emplacement des différents fichiers. Chacun de ces chemins est renseigné grâce à un préfixe (`--images`, `--ead_eac`, `--alto`, `--rng`). L'usager donne également un nom au fichier XML qu'il souhaite obtenir en sortie, préfixé (`--output`) ;
2. Le script principal charge et stocke distinctement les différents fichiers à partir de l'extension de ces derniers (le script fait un appel au module `extract_utils.py` via le *package built-in glob* qui gère la recherche de chemins de style Unix) ;
3. Une étape préliminaire contrôle la présence des fichiers, en cas d'absence de ces derniers le programme envoie un message (*log*) d'erreur et interrompt le programme en cours ;
4. La première étape du traitement consiste à créer des fichiers XML « catalogues » qui regroupent les fichiers XML fournis en entrée (module `build_utils.py`). Les fichiers XML ALTO sont disposés dans un premier catalogue XML (`catalog_alto.xml`) et les fichiers XML EAD-EAC suivent la même procédure (`catalog_ead_eac.xml`).

4.5 Simuler l'agrégation des données et la validation d'un fichier XML-TEI pivot avec un script Python

Nous verrons dans la section suivante l'utilité de ces deux fichiers XML pour les transformation XSL ;

5. L'étape suivante consiste à extraire des fichiers, dans les catalogues XML. Les données sont placées dans deux nouvelles arborescence XML distinctes qui correspondent aux informations des fichiers XML ALTO placés eux dans une première structure arborescente réparti sur deux niveaux : <**facsimile**> et <**text**>. Les informations des XML EAD et XML EAC-CPF sont disposés dans un <**teiHeader**>, soit une seconde structure arborescente. Les structures sont créées à partir des feuilles XSL, qui sont lues par le pré-processeur XSLT *Saxon*. Ce dernier est appelé par une ligne de commande qui s'exécute automatiquement durant le programme par l'intermédiaire d'une fonction du *package* OS qui permet d'interagir avec le système d'exploitation ;
 6. Les deux structures sont alors rassemblées (*merge*) en une seule qui comprend désormais le <**teiHeader**>, le <**facsimile**> et le <**text**> ;
 7. Les métadonnées EXIF sont extraites des images et rassemblées dans une arborescence <**xenoData**> qui rejoint l'arborescence précédente, de manière à constituer le fichier XML-TEI complet. À noter que durant cette étape certaines valeurs EXIF n'ont pas été décodées au moment de leurs insertions dans l'arborescence. Nous avons donc ajouté une valeur par défaut « *bytes values* », pour celles qui ne sont pas prises en compte⁴⁴ ;
 8. Le programme génère alors une sortie XML comme étant le nouveau fichier XML TEI et le sauvegarde dans le dossier par défaut **Output** ;
 9. Si l'usager a spécifié une option de validation (**--rng**) et renseigné un schéma de validation, le programme appelle le module **validation_utils.py** qui vérifie la conformité XML et la validation du XML TEI⁴⁵. Le programme s'interrompt après avoir renvoyé un message d'erreur ou de succès à l'usager.

```
+---+---+---+---+---+ +---+---+---+---+---+  
| L | e | c | t | a | u | r | e | p | | T | E | I | | G | e | n | e | r | a | t | o | r |  
+---+---+---+---+---+ +---+---+---+---+---+  
  
Build XML catalog for ead_eac files in progress...: 100% | 4/4 [00:00<00:00, 4807.23it/s]  
Build XML catalog for alto files in progress...: 100% | 21/21 [00:00<00:00, 8325.97it/s]  
Extraction and assembly of EXIF metadata in progress...: 100% | 21/21 [00:00<00:00, 146.60it/s]  
  
TEI file generated : correct_legay  
Schema Validation in progress...  
Great Job ! Your document is valid !  
(tei-gen) (base) macbook-pro-de-lucas:generator_Lectaurep2TEI lucasterriel$
```

FIGURE 4.9 – Capture du terminal qui montre le bon déroulement du CLI Generator Lectaurep-TEI. ©L. Terriel, 2020, Diagrams.net

44. Une balise `<exif>` doit obligatoirement contenir une valeur. Dans le canevas XML-TEI j'ai ajouté au niveau du `<encodingDesc>` cette information de manière à ce que l'on puisse l'interpréter ultérieurement.

45. Pendant les tests nous avons utilisé un schéma RelaxNG TEI ALL.

4.5.2 Deux feuilles de styles XSLT pour obtenir des arborescences TEI

Nous avons esquissé, plus haut, durant l'exécution du programme le rôle des feuilles de transformation XSL : `Lectaurep_ALT02TEI.xsl` et `Lectaurep_EADEAC2TEI.xsl`. Pour rappel une transformation XSL se base sur le langage XSLT qui est une syntaxe XML permettant de spécifier, suivant des règles de transformation, de quelle manière un ou des fichier(s) XML doi(ven)t être transformé(s) en un autre document (Cf. Figure 4.10). Il s'agit en somme d'un langage de réécriture d'arbres. Une transformation s'effectue à l'aide d'un pré-processeur (comme *Saxon* ou *Xalan* écrits tous les deux en langage Java) et peut s'exécuter dans un éditeur XML comme *Oxygen XML Editor* ou bien en ligne de commande comme dans le programme *Generator Lectaurep-TEI*.

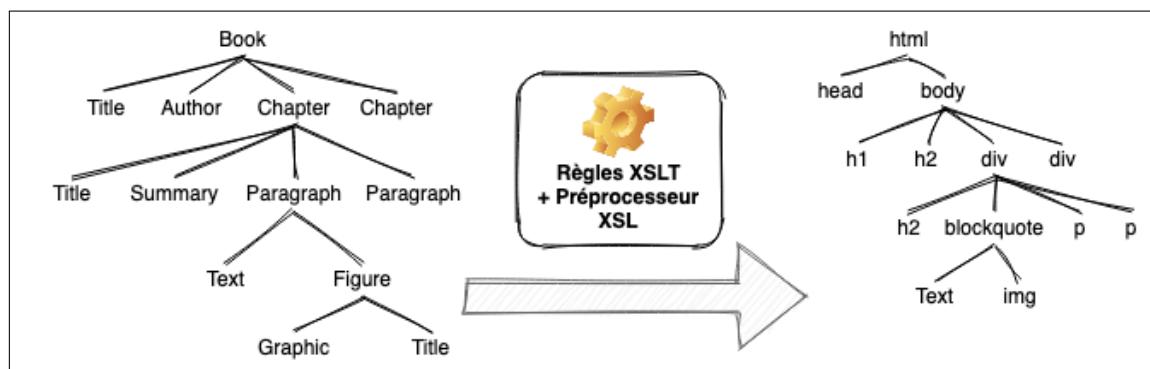


FIGURE 4.10 – Illustration d'une transformation XSLT d'un document XML vers un document HTML ©L. Terriel, 2020, Diagrams.net

4.5 Simuler l'agrégation des données et la validation d'un fichier XML-TEI pivot avec un script Python

Les feuilles de style XSL créées pour le programme sont les suivantes :

- `Lectaurep_ALTO2TEI.xsl` correspond à la transformation des fichiers XML ALTO vers des structures TEI `<facsimile>` et `<text>`. Ce fichier a été repris et adapté sur la base de la feuille XSL proposée par le Dutch Language Institute (INL)⁴⁶. Certaines règles de transformation ont dû être modifiées pour correspondre aux balises spécifiques des fichiers XML ALTO de Lectaurep. De plus, une fonction XSLT a dû être implémentée en sus pour permettre de concaténer les points de coordonnées de l'élément ALTO `<Polygon>` sous la forme de coordonnées avec un séparateur $(x,y\ x,y\ x,y\dots)$ pour être inclus dans l'attribut `@points` de l'élément TEI `<zone type="Polygon">`⁴⁷, sans cela le document TEI ne pouvait être validé.
- `Lectaurep_EADEAC2TEI.xsl` correspond à la transformation des éléments XML EAD-EAC vers une structure `<teiHeader>`. Elle a été créée *ex-nihilo* pour coller au mieux à la structure d'accueil présentée dans le canevas XML-TEI de Lectaurep.

Ces feuilles XSL s'appliquent à plusieurs fichiers en même temps (quatre fichiers XML EAD-EAC et vingt-et-un fichiers XML ALTO pour les tests). J'ai dû trouver un moyen de parcourir l'ensemble de ces fichiers en une seule fois pour en extraire les informations vers le nouveau document TEI. Une solution consiste à recourir à la fonction XPath⁴⁸ `collection()` dans les feuilles XSL⁴⁹. Son utilisation requiert alors de constituer au préalable un catalogue regroupant les fichiers XML à transformer (Cf. Figure 4.11). La fonction est placée dans la feuille XSL avec le nom du fichier XML « catalogue » à transformer.

À partir de là, les règles de transformation seront effectives sur l'ensemble des documents XML contenus dans ce catalogue et présentant le même sous-ensemble localisé par la requête XPath.

46. Dutch Language Institute (INL), *alto2tei...*

47. suivre l'échange du 24 août 2020 sur *stackoverflow*, URL : <https://stackoverflow.com/questions/63569657/xslt-concatenate-a-list-of-numbers-with-a-separator-according-to-a-pattern>

48. **XPath** est un langage de requête utilisé pour parcourir ou localiser une portion d'un document XML. XSLT utilise XPath dans ses règles de transformation pour repérer des éléments à modifier.

49. Pour plus de précision sur l'utilisation de la fonction XPath `collection()` consulter Martin Holmes, *The XPath collection() function : pulling in multiple documents*, 2013, URL : http://web.uvic.ca/~mholmes/dhoxss2013/handouts/collection_function.pdf

```
1 <collection>
2   <doc href=".//FRAN_IR_051379.xml"/>
3   <doc href=".//FRAN_NP_011490.xml"/>
4   <doc href=".//FRAN_IR_041698.xml"/>
5   <doc href=".//FRAN_NP_010150.xml"/>
6 </collection>
```

FIGURE 4.11 – Exemple d'un fichier XML « catalogue » qui regroupe des fichiers XML EAD-EAC ©L. Terriel, 2020

4.5.3 Perspectives d'évolution pour le *Generator Lectaurep-TEI*

Il n'a pas été prévu, pour le moment, d'apporter des modifications majeures à l'outil. En effet, celui-ci suit de près les évolutions du canevas XML-TEI dans la continuité du projet. Quand le schéma et les réflexions de Lectaurep autour du canevas seront plus développés, il serait envisageable, grâce à ce petit programme de générer un fichier XML-TEI et de tester son intégration dans *eScriptorium* ou dans des applications *web* pour des visualisations des données encodées en TEI (visualisation du texte et de l'image du répertoire dans une interface et des métadonnées associées). Enfin, les différents modules développés pour ce *script*, comme le validateur RelaxNG (*validation_utils.py*) ainsi que les feuilles de transformation XSL peuvent être réutilisées indépendamment de l'outil dans d'autres types de projets par ALMAnaCH.

Troisième partie

**Développer une application en
Python pour évaluer les modèles
HTR**

La deuxième partie du stage était axée sur une mission de développement applicatif.

Au sein d'ALMAñaCH, il s'agissait de disposer d'un outil interne permettant de visualiser rapidement des informations sur la qualité des transcriptions réalisées par les modèles HTR entraînés avec le CLI *Kraken*.

Cependant, l'outil devait être généralisable, pour être réutilisé dans d'autres projets. Il était souhaitable, également, que l'application prenne en ligne de compte la possibilité, à terme, d'interagir avec le fichier XML-TEI pivot pour importer les transcriptions HTR et les vérités terrains d'*eScriptorium* et récupérer les métriques d'évaluation.

L'application *Kraken-Benchmark*, créée durant mon stage, a permis de réaliser en partie ces objectifs.⁵⁰.

50. La seconde mission de stage a été présentée sous la forme d'une *issue* GitLab accessible via le lien : <https://gitlab.inria.fr/dh-projects/kraken-benchmark/-/issues/1> [à titre informatif, un accès peut-être requis]

Chapitre 5

État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken

5.1 Banc d'essai des outils existants : limites et avantages

Dans de nombreux projets de recherche et de développement d'outils informatiques, on démarre rarement d'une simple page blanche. Avant de me pencher directement sur les aspects de développement et de gestion du projet, j'ai cherché à faire le bilan des solutions qui existaient pour Lectaurep jusqu'à maintenant pour évaluer les transcriptions.

Lorsque Lectaurep utilisait encore *Transkribus*, les résultats étaient obtenus au cours d'une longue procédure. Une fois que le nombre de pages transcris était suffisant, un mail était envoyé aux membres de l'équipe *Transkribus* pour leur signaler qu'ils pouvaient entraîner le modèle, en spécifiant les pages à utiliser en guise de vérités terrains. Une fois cette tâche réalisée, un échange sur plusieurs mails s'engageait avec l'équipe de *Transkribus* proposant parfois de compléter les données avec d'autres transcriptions similaires. Pour évaluer le modèle, on pouvait accéder à un rapport contenant le taux d'erreur par caractères (*Character Error Rate* ou CER) et le taux d'erreur par mots (*Word Error Rate* ou WER) via un onglet « Tools » puis l'onglet « Choose ... »¹. Nous reviendrons sur la définition de ces métriques dans la section suivante. Toujours par le biais de l'onglet « Tools » puis de l'onglet « Compare Text Versions ... », on pouvait accéder aux deux versions du texte superposées, comprenant le document de vérité terrain transcrit manuellement et la prédiction obtenue par le modèle².

1. Cf. Annexes, Figure E.1

2. Cf Annexes, Figure E.2 et voir M.L. Bonhomme, *Défis et opportunités de la reconnaissance*

CHAPITRE 5 : État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken

Kraken est l'outil privilégié pour l'entraînement des modèles de segmentation et de transcription ainsi que pour effectuer des prédictions à partir des modèles. Cependant concernant, l'étape d'évaluation de la transcription obtenue, l'affichage dans le terminal ne permet pas de bien discerner les erreurs, du moins de bien les localiser dans le texte.

En effet, le rapport d'évaluation (Cf. Figure 5.1) de la transcription s'obtient grâce à la commande `$ ketos test -m model images` qui permet d'obtenir : le nombre total de caractères dans la vérité terrain, un taux de réussite général qui correspond au CER, le nombre d'insertions, de suppressions et de substitutions de caractères ainsi qu'une liste des erreurs les plus fréquentes.

```
== report ==
35619 Characters
336 Errors
99.06% Accuracy

157 Insertions
81 Deletions
98 Substitutions

Count Missed %Right
27046 143 99.47% Syriac
7015 52 99.26% Common
1558 60 96.15% Inherited

Errors Correct-Generated
25 { } - { COMBINING DOT BELOW }
25 { COMBINING DOT BELOW } - { }
15 { . } - { }
15 { COMBINING DIAERESIS } - { }
12 { } - { }
10 { } - { . }
8 { COMBINING DOT ABOVE } - { }
8 { , } - { }
7 { ZERO WIDTH NO-BREAK SPACE } - { }
```

FIGURE 5.1 – Exemple de rapport fourni par kraken ©2015, *Kraken API*, URL : <http://kraken.re/api.html>

J'ai également voulu vérifier si d'autres outils d'évaluation similaires existaient déjà en parcourant les dépôts de code sur la plate-forme *Github*. J'ai donc relevé trois outils parmi lesquels : *Werpp*³, *XER*⁴ et *WER-in-python*⁵ à partir de deux fichiers texte brut d'exemples comprenant 10 à 20 mots.

Le premier représentant un document de vérité terrain, l'autre étant censé montrer la transcription avec des erreurs volontairement insérées (omission de la ponctuation, abus de majuscules, espaces conséquences entre certains mots etc.).

automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris..., pp. 41

3. *Werpp*, URL : <https://github.com/nsmartinez/WERpp>

4. *XER*, URL : <https://github.com/jpuigcerver/xer>

5. *WER-in-python*, URL : <https://github.com/zszyellow/WER-in-python>

5.1 Banc d'essai des outils existants : limites et avantages

Une fois les tests effectués, j'ai pu relever les points suivants :

- Certains programmes n'utilisaient pas les mêmes versions de Python, comme *Werpp* qui présentait des défauts de compatibilité avec la version 3. Pour le faire fonctionner, il a fallu modifier quelques lignes de codes afin de le rendre exécutable ;
- Certaines options proposées par ces programmes comme dans *Werpp* ou *XER* ne fonctionnaient pas. Ainsi dans le cas de *XER* le chargement des fichiers au format texte ne fonctionnait pas. Il fallait inscrire les phrases dans le terminal avec une commande du type `$ xer -i str -r "Je suis un document valide" -t "je suis Unn DoCument, invalide!"` ce qui n'est évidemment pas du tout envisageable pour des transcriptions de plus de 10 lignes de texte. D'autres options comme la colorisation des résultats pour les lettres manquantes, proposé par *Werpp*, ne fonctionnaient pas en raison de l'obsolescence des *packages* requis ;
- Certains outils ne permettaient pas de cumuler en sortie plusieurs résultats en même temps comme le WER et le CER. C'est par exemple le cas de *WER-in-python* et de *Werpp* ;
- Enfin les temps de calculs pouvaient s'avérer très longs et la plupart des codes n'étaient pas bien documentés, l'usage des commandes devant être déduites de la lecture du code.

Si la plupart de ces outils ne se sont pas révélés directement utilisables pour notre projet, pour des utilisations fréquentes, ce premier état de l'art m'a permis de me poser des questions quant aux fonctionnalités qui pouvaient être réutilisées et aux métriques qui pourraient s'avérer utiles dans la future application.

De plus la lecture du code, m'a permis de me familiariser avec la création de CLI par le biais du *package* intégré à Python *argparse*.

Argparse est un *package* Python permettant d'écrire rapidement des programmes sous forme de CLI en récupérant un ensemble de fonctions réutilisables pour permettre entre autre, la gestion des arguments entrés par l'utilisateur dans le terminal, la documentation, etc.

Afin de me familiariser avec les logiques de fonctionnement du CLI, j'ai créé un premier programme `cerwer_tool`⁶ en adaptant certaines fonctions des outils décrits plus haut, qui allait préfigurer le développement de l'outil *Kraken-Benchmark* spécifique au projet.

6. Le dépôt du code de l'outil `cerwer_tool.py` est disponible sur *Github*, URL : https://github.com/Lucaterre/cerwer_tool

CHAPITRE 5 : État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken

L'outil `cerwer_tool.py` permettait, en outre, au moyen d'une ligne de commande prenant en argument deux fichiers texte correspondant à une vérité terrain et une prédition, d'émettre un rapport dans le terminal comportant le nom de l'utilisatrice ou de l'utilisateur, la date, le texte de référence, le texte prédit, de comparer le nombre de mots et de lettres insérées, substituées et supprimées et enfin d'afficher le WER et le CER.

L'utilisatrice ou l'utilisateur avait la possibilité de sauvegarder le rapport dans un fichier texte pour en garder une trace et de produire un graphique présentant le nombre d'inscriptions, substitutions et de suppressions sur le nombre de mots total (Cf. Figure 5.2).

Cependant cet outil avait des limites : il supposait de récupérer en amont la transcription HTR, avec *Kraken*, dans un fichier texte pour utiliser `cerwer_tool.py`. Comme nous le verrons par la suite, j'ai décidé d'intégrer cette chaîne de traitement HTR directement dans l'outil *Kraken-Benchmark* en réutilisant certaines parties du code de *Kraken* par l'intermédiaire de son API.

```
Entrer votre prénom et votre nom : Lucas Terriel
Désirez-vous obtenir un graphique ? [0/n] : n
*****
\--- RAPPORT ---
* Effectué par : Lucas Terriel
* Date - Heure : 2020-04-23 14:33:40.218037
* Temps d'exécution : 1.478134 secondes ---

Phrase de référence      :
Pruvot (par Julien) à Paris, rue Crozatier 6 et autres, consorts : en blanc, pour toucher !

Phrase du modèle         :
Pruvot (par Julien) Paris, rue 6 et autres, : en blanc, pour toucher !

Phrase de référence tokenisée :
['Pruvot', '(', 'par', 'Julien', ')', 'à', 'Paris', ',', 'rue', 'Crozatier', '6', 'et', 'autres', ',', 'consorts', ':', 'en', 'blanc', ',', 'pour', 'toucher', '!'] Longeur : 22 token(s)

Phrase du modèle tokénisée :
['Pruvot', '(', 'par', 'Julien', ')', 'Paris', ',', 'rue', '6', 'et', 'autres', ',', ':', 'en', 'blanc', ',', 'pour', 'toucher', '!'] Longeur : 19 token(s)

Mots : Ins : 0, Subs : 0, Dels : 3
Lettres : Ins : 0, Subs : 0, Dels : 18
Résultats du WER (en %)   : 13.64%
Résultats du WER          : 0.136
Résultats du CER (en %)   : 24.00%
Résultats du CER          : 0.240
*****
```

FIGURE 5.2 – Capture d'écran du rapport produit par le prototype `cerwer_tool.py` ©L. Terriel, 2020, Pycharm

5.2 Des métriques pour comparer la transcription automatique et la vérité terrain

Après avoir formulé un état de l'art, je me suis penché sur les méthodes de calcul qui permettent d'évaluer une transcription HTR avec sa vérité terrain. C'est un problème qui peut être ramené à la comparaison de deux chaînes de caractères entre elles (comparaison *text-to-text*). Dans un deuxième temps, pour permettre d'évaluer les transcriptions dans un contexte de traitement par le TAL, je me suis également intéressé aux métriques permettant d'apprécier la proximité sémantique entre des mots appartenant à deux documents différents.

J'ai exposé et synthétisé ce travail de recherche dans un *notebook* Jupyter. Un *notebook* peut être conçu comme un calepin électronique qui permet d'écrire du texte brut et du code au même endroit. Comme nous le verrons par la suite, il est indispensable dans tout travaux concernant le traitement de données en ML et DL, notamment pour expérimenter des calculs, des *scripts* et de préciser certaines étapes⁷. Le *notebook* résume brièvement les définitions et les usages des métriques, les visualisations possibles à partir de ces dernières et les différents algorithmes qui permettent de les implémenter dans un programme.

5.2.1 La comparaison de chaînes de caractères

Parmi les métriques et les algorithmes associés qui permettent d'évaluer la similarité syntaxique entre deux phrases :

Le similarité Ratcliff/Obershelp - L'algorithme de Ratcliff/Obershelp (ou *Gestalt Pattern Matching*) se base sur la recherche de sous-chaînes de caractères communes (*subpattern*) entre deux séquences de caractères.

Le principe de l'algorithme de Ratcliff/Obershelp repose sur une découpe des phrases en deux parties en se plaçant par rapport à une ancre qui correspond au premier point de différentiation entre les deux séquences. De part et d'autre, le processus itératif évalue à gauche puis à droite des séquences les plus longues les sous-chaînes communes, jusqu'à ce que la longueur des séquences soit intégralement parcourue.

7. Le *notebook* de recherche est disponible dans plusieurs formats dans les Annexes C, /C-Application_Kraken_Benchmark/Documentation-Reasearch/
Evaluation de la similarité entre deux séquences dans le contexte de la reconnaissance automatique de caractère

CHAPITRE 5 : État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken

La similarité Ratcliff/Obershelp ($Similarite_{RO}$) calcule deux fois le nombre de caractères effectivement reconnus ($Sub_{caracteres}$) dans les sous-chaînes les plus longues ($2 \cdot Sub_{caracteres}$) sur le nombre total de caractères compris dans les deux phrases (signe typographique et espace compris).

soit :

$$Similarite_{RO} = \frac{2 \cdot Sub_{caracteres}}{LongChaine_1 + LongChaine_2}$$

où :

$$0 \leq Similarite_{RO} \leq 1$$

Par exemple, soit $S_1 = \langle\langle EN L'AN 1920 PAR LA PROCURATION \rangle\rangle$ et $S_2 = \langle\langle EN L'AN 1920 PAR LE PROCUREUR \rangle\rangle$

pour

$$Similarite_{RO}(S_1, S_2) = \frac{2 \cdot ([EN\ L'\ AN\ 1920\ PAR\ L] + [PROCUR])}{S_1 + S_2} = \frac{50}{60} \simeq 0,83$$

Dans cet exemple, l'algorithme de Ratcliff/Obershelp trouve dans un premier temps « EN L'AN 1920 PAR L » à gauche comme la plus longue sous-chaîne commune puis à droite « PROCUR » (espace à gauche compris)⁸.

Pour inclure ces métriques dans un programme informatique le *package* Python *difflib* permet de récupérer des fonctions qui se basent sur cet algorithme.

L'avantage est qu'il permet d'obtenir un score rapidement, afin de comparer deux chaînes de caractères. Cependant elles ne rendent pas compte précisément des modifications de caractères qui ont pu s'opérer lors du passage d'une chaîne à une autre. Nous allons voir en quoi les distances mathématiques peuvent nous y aider.

8. Exemple d'implémentation dans un *script* Python Cf. Annexes F, Figure F.3

5.2 Des métriques pour comparer la transcription automatique et la vérité terrain

Les distances de Levenshtein, de Hamming et de Damerau-Levenshtein

- La distance de Levenshtein⁹ (équivalent de la distance d'édition), est une distance mathématique, et une généralisation de la distance de Hamming, dans le sens où la première peut travailler sur des chaînes de longueurs différentes, mais pas dans le cas de la seconde.

Cette distance évalue le coût minimal de transformation d'une chaîne de caractères R en une chaîne de caractère P en effectuant les opérations suivantes auxquelles sont associés un coût de 1 :

- La **substitution** d'un caractère de R par un caractère de P ;
- L'**insertion** d'un caractère dans R par P ;
- La **suppression** d'un caractère dans R par P ;

Pour illustrer le calcul de cette distance nous pouvons prendre les exemples suivants :

- soit $R = \text{magasin}$ et $P = \text{magasin}$, alors $LD(R, P) = 0$, car il n'y a pas de changement.
- soit $R = \text{magasin}$ et $P = \text{megasinier}$, alors $LD(R, P) = 4$, car il y a 3 insertions « ier » et une substitution « a » en « e ».

La distance de Damerau-Levenshtein¹⁰, quant à elle, est une variation de la distance de Levenshtein qui inclut les transpositions (échange de deux lettres) dans les opérations (pour prendre en compte les fautes d'orthographe ou les échanges de caractères).

Par exemple dans le cas des mots « acceuil », « auteuil » et « accueil », la distance de Levenshtein calcule une distance de 2 entre chaque mots. Cependant « acceuil » et « accueil » sont plus proches (transposition de « eu » en « ue ») que « auteuil » et « acceuil », la distance de Damerau-Levenshtein prend en compte cette différence. Ainsi la modification « acceuil » et « accueil » sera pondérée à 1 tandis que la modification « auteuil » et « acceuil » sera pondérée à 2. Une transposition de deux caractères coûte moins qu'une substitution.

Cette distance est plutôt utilisée dans le cadre de correcteur orthographique, de plus la complexité algorithmique de la distance de Damerau-Levenshtein est plus élevée que la distance de Levenshtein, ce qui est une caractéristique à prendre compte durant l'implémentation. La distance de Damerau-Levenshtein sera plus longue à calculer que la distance de Levenshtein (environ 20 secondes pour la première et 0,01 secondes pour la seconde (implémentation la plus rapide) pour des chaînes d'environ 350 caractères)¹¹.

9. Établie en 1965 par le mathématicien russe Vladimir Levenshtein (1935-2017)

10. François-Régis Chaumartin et Pirmin Lemberger, *Le traitement automatique des langues. Comprendre les textes grâce à l'intelligence artificielle*, Dunod, 2020, pp. 132

11. Exemple d'implémentation dans un *script* Python Cf. Annexes F, Figure F.3

Il existe plusieurs algorithmes pour implémenter cette distance¹²; la tâche d'implémentation peut être facilité par des *packages* en Python, comme la fonction `_fast_levenshtein()` de l'API de l'*Kraken*¹³ ou encore *python-levenshtein*¹⁴ qui est une implantation plus rapide de la distance (extension écrite en langage C).

Nous allons voir que cette distance est importante si on veut pouvoir calculer le taux d'erreur par caractères (CER) et le taux d'erreur par mots (WER).

Le taux d'erreur de caractères et le taux d'erreur de mots - Le taux d'erreur de caractères (*Character error rate* ou CER) et le taux d'erreur de mots (*Word error rate* ou WER) sont des taux d'erreurs très utilisés pour constater l'efficacité d'un modèle de reconnaissance. Le CER et le WER se calcule de la manière suivante :

Si :

- N est le nombre total de caractères ou de mots contenus dans la phrase de référence ;
- S (*Substitution*) est le nombre de substitutions (caractères ou mots incorrectement reconnus) ;
- D (*Deletion*) est le nombre de suppressions (caractères ou mots omis) ;
- I (*Insertion*) est le nombre d'insertions (caractères ou mots ajoutés).

Le CER se calcule tel que :

$$\text{Character Error Rate} = \frac{S + D + I}{N_{\text{total de caractères}}}$$

Le WER se calcule tel que :

$$\text{Word Error Rate} = \frac{S + D + I}{N_{\text{total de mots}}}$$

Dès lors si on connaît la distance de Levenshtein D pour une phrase de référence notée R et une phrase de prédiction H , le CER peut être ramené à l'expression suivante :

$$CER = \frac{D(R, H)}{N_{\text{total de caractères}}}$$

et le WER :

$$WER = \frac{D(R, H)}{N_{\text{total de mots}}}$$

12. Wikibooks, *Algorithm Implementation/Strings/Levenshtein distance*, 2020, URL : https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance (visité le 17/09/2020)

13. *Kraken API...*

14. *python-Levenshtein 0.12.0*, URL : <https://rawgit.com/ztane/python-Levenshtein/master/docs/Levenshtein.html>

5.2 Des métriques pour comparer la transcription automatique et la vérité terrain

Les résultats obtenus sont interprétés de la sorte : plus le taux approche 0 plus la reconnaissance est efficace, à l'inverse, plus le taux se rapproche de 1, plus le texte a eu du mal à être reconnu. Il peut même dépasser 1, si la reconnaissance est très mauvaise, surtout s'il y a eu beaucoup d'insertions. Nous avons été confronté à ce cas, comme nous le verrons en chapitre 7. Notons enfin qu'il est possible de ramener ces taux d'erreurs à des pourcentages pour une meilleure visibilité.

A partir du WER on peut également calculer le taux de reconnaissance de mots (ou *Word Accuracy* (WAcc)), calculé pour obtenir un pourcentage de la manière suivante (à noter que le résultat peut-être négatif si le WER dépasse 1) :

$$W_{ACC} = (1 - WER) \cdot 100$$

Le CER est plus significatif que le WER et que le taux de reconnaissance par mots dans le sens où ces derniers sont corrélés aux erreurs de prédiction des caractères effectués par le modèle. *De facto* le WER augmente et le taux de reconnaissance par mots baisse quand le CER augmente. Cependant, dans un projet HTR ces métriques peuvent se compléter ; ainsi il est intéressant de savoir si un modèle qui fait beaucoup de fautes épargne les mots et à l'inverse si un modèle fait peu de fautes, ci ces dernières se retrouvent sur tous les mots¹⁵.

5.2.2 Estimer la similarité entre deux documents

L'intuition est de prendre la vérité terrain comme premier document et sa prédiction par le modèle HTR comme un deuxième document. L'indice de Jaccard ainsi que la similarité cosinus sont des métriques qui évaluent la proximité entre des textes :

L'indice de Jaccard - Son but est d'estimer le pourcentage de mots communs aux deux documents. L'indice évalue, sur l'union de deux ensembles (deux phrases par exemple), la taille de l'intersection qui regroupe les mots commun à la transcription de référence et à la transcription HTR, qu'il divise par la taille de l'union de deux ensembles.

L'indice est formulé ainsi :

$$J(doc_1, doc_2) = \frac{|doc_1 \cap doc_2|}{|doc_1 \cup doc_2|}$$

Plus le résultat approche de 1, plus les documents sont similaires, car le nombre des mots partagés par la référence et par sa transcription HTR dans l'intersection est important. À l'inverse, si le résultat est plus proche de 0, alors on peut en conclure que

15. Exemple d'implémentation dans un *script* Python Cf. Annexes F, Figure F.3

CHAPITRE 5 : État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken

les deux textes sont davantage éloignés l'un de l'autre.

La similarité cosinus - Si l'on reprend la Figure 2.13 à la section 2.3, il s'agit de calculer l'angle cosinus noté $\cos(\theta)$ entre des mots, alors représentés sous la forme de vecteurs, afin de mesurer leur degré de proximité. La formule de la similarité cosinus est la suivante :

$$\text{similarité cosinus} = \cos(\theta) = \frac{\vec{V1} \cdot \vec{V2}}{\|\vec{V1}\| \cdot \|\vec{V2}\|}$$

où :

$$\|\vec{V1}\| \text{ et } \|\vec{V2}\| > 0$$

Les résultats obtenus sont interprétés de la manière suivante, en fonction de la valeur de $\cos(\theta)$:

- Plus elle tend vers 0, plus les vecteurs sont dits indépendants ou opposés (orthogonaux) : les documents sont donc éloignés ;
- Plus elle tend vers 1, plus les vecteurs sont dits (colinéaires de coefficient positif) donc les documents sont proches ;

Pour illustrer cette métrique prenons un exemple concret issu d'un répertoire de notaire. En nous appuyant sur une vérité terrain (R pour référence) et une prédiction (P) réalisées par un modèle de transcription qui a émis quelques erreurs.

R = « An 1920, mois d'Avril pour gerer 2 maisons à Paris Procuration, par Rosalie Adélaïde Eugénie Isoarel, dt à Paris, rue de la Collégiale »

P = « sAn 19s2iOt, mois de Avril per gerer 2ts maisons à Paris Procuration, par Reselie Adelaide Eugenie lisrel, dt a Peres, rue de la Collégiale »

Leurs termes sont : « an, mois, avril, gerer, maisons, paris, procurement, roasalie, adélaïde, reselie, adelaide, eugénie, eugenie, isoarel, isrel, rue, collegiale »

On pourra alors représenter R et P sous forme de vecteurs à 17 dimensions (correspondant au nombre de termes relevés) dont les valeurs numériques représentent les occurrences d'apparition du mot dans la phrase :

$$R = [1, 1, 1, 1, 1, 2, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1]$$

$$P = [0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1]$$

5.2 Des métriques pour comparer la transcription automatique et la vérité terrain

Si la norme du vecteur R se calcule tel que :

$$\|\vec{R}\| = \sqrt{R_1 \cdot R_1 + R_2 \cdot R_2 + \dots + R_n \cdot R_n}$$

On aura donc $\|\vec{R}\| = 4.0$ et $\|\vec{P}\| = 3.46$

Le produit scalaire de $R \cdot P$ se calcule tel que :

$$R \cdot P = R_1 \cdot P_1 + R_2 \cdot P_2 + \dots + R_n \cdot P_n$$

On aura donc $R \cdot P = 9$

Dès lors, si la formule de la similarité cosinus s'exprime tel que :

$$\text{SimCos}(R, P) = \frac{R \cdot P}{\|\vec{R}\| \cdot \|\vec{P}\|}$$

On obtient alors le résultat suivant : $\text{SimCos}(R, P) = 9 / (4.0 \cdot 3.46) = 0.65$

Les documents obtiennent un score de similarité de 65% dès lors le modèle à commis des erreurs.

5.2.3 Remarques complémentaires sur les métriques d'évaluation utilisées

Au vue des différentes métriques présentées, j'ai décidé de travailler sur deux types de métriques dans l'application *Kraken-Benchmark* afin d'évaluer la correspondance entre deux phrases.

Une première catégorie de métriques, de type « similarité sémantique », empruntées à la fouille de textes, va chercher à savoir si les mots généraux ou les entités nommées (lieux, personnes ou types d'actes) sont équivalents dans les deux textes et sont donc bien reconnus par le modèle. C'est le cas de la similarité cosinus et de l'indice de Jaccard. Cependant, il s'agit d'une utilisation non conventionnelle dans le cadre de l'évaluation HTR de ces métriques qui évaluent la proximité sémantique d'un grand ensemble de documents. De plus, elles ne seront pas assez précises pour savoir si les deux phrases sont strictement équivalentes et syntaxiquement correctes car pour se calculer ces métriques nécessitent souvent qu'on enlève les mots fréquents (*stop words*) qui peuvent également contenir des fautes de transcription (qui sont intéressants à relever si l'on veut obtenir une bonne qualité de transcription) pour réaliser une représentation de type « sac de mots » (*bag of words*) et obtenir un modèle vectoriel des textes (*word embedding*).

Les métriques, dites de « similarité syntaxique », travaillent, quant à elles, sur la position exacte des mots et des caractères dans la phrase. Elles nous apporteront des précisions sur les types de modifications qui ont été le plus souvent opérées par le modèle. Il s'agira des métriques « classiques » de l'HTR comme la distance de Levenshtein, le CER

CHAPITRE 5 : *État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken*

et le WER.

Il faut noter que pour permettre l'implémentation de ces métriques dans un programme informatique des étapes de normalisation des données textuelles sont requises : on parle de *data pre-processing*. On a donc eu recours durant le développement de l'application à des fonctions de découpage des phrases en caractères ou en mots (*tokenisation*), de suppression des mots les plus fréquents (*stops words*), ainsi que des moyens de convertir les mots sous formes de valeurs numériques, comme la transformation en vecteurs (*word embedding*).

Pour terminer cette section, il est possible de retrouver dans le *notebook* intitulé « Évaluation de la similarité entre deux séquences dans le contexte de la reconnaissance automatique de caractères » davantage de précisions sur les métriques, les étapes de normalisation du texte, leurs différentes implémentations en Python, des tests de durée d'exécution et les visualisations réalisées (graphiques, matrices de confusion etc.) (Cf. Annexes C).

Chapitre 6

Le développement d'une application : Kraken-Benchmark

6.1 Modélisation

6.1.1 Les objectifs fixés au début du développement

En concertation avec ma tutrice de stage, Alix Chagué, j'ai fixé des cas d'usages (*use case*) et formalisé les objectifs idéaux à atteindre dans la conception de l'application. Ceci dans l'optique de décrire les exigences fonctionnelles de la future application. Nous avons donc relevé que :

- L'application doit pouvoir comparer et évaluer la qualité d'un modèle de transcription et/ou de segmentation à l'échelle d'une ou de plusieurs images ;
- Les résultats de l'évaluation doivent être accessibles et visualisables dans une interface conviviale pour l'utilisateur (*user-friendly*) ;
- Le rapport produit doit pouvoir être exporté dans plusieurs formats ;
- L'utilisateur doit avoir la possibilité de charger les images et le modèle facilement ;
- Le programme doit être écrit en Python 3 pour assurer sa maintenabilité dans le temps ;
- Le code, et l'application, doivent disposer d'une documentation interne et externe ;
- Le programme doit pouvoir être généralisé à d'autres projets et/ou viser une interopérabilité avec la plate-forme *eScriptorium* à terme ;

Nous verrons dans la section 6.3 les objectifs qui ont été atteints, ceux qui restent en cours de développement et ceux qui ont émergés au cours des tests et rédaction du code.

6.1.2 Un écosystème Python orienté pour la science des données et la conception d'application web

Pour réaliser l'application *Kraken-Benchmark* en Python¹, il m'a fallut disposer de solutions pour concevoir l'interface, mais aussi pour effectuer des tâches plus précises comme les calculs et les représentations graphiques de données (*data visualisation*). Quand il s'agit d'évaluer des données, Python fait graviter autour de lui un ensemble d'outils orientés pour la science des données (*data science*, Cf. Figure 6.1) comprenant la manipulation et la fouille de données, des statistiques, des visualisations, l'application de modèles mathématiques, et l'apprentissage automatique.

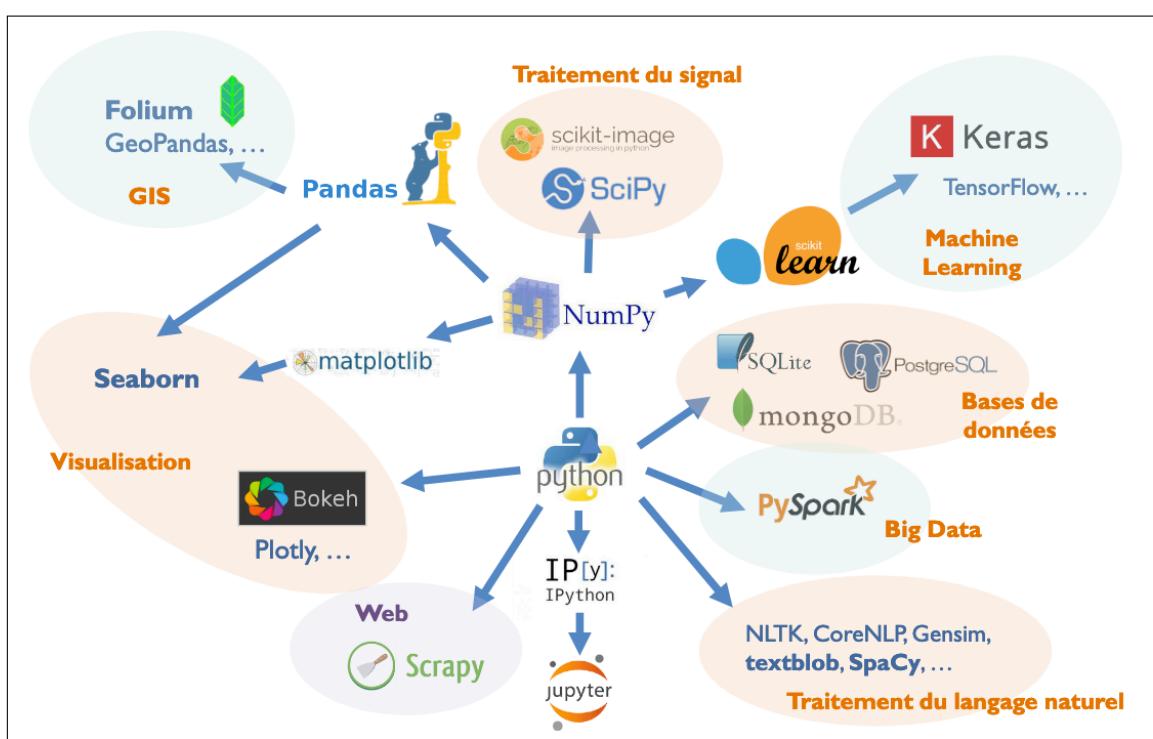


FIGURE 6.1 – L'écosystème Python pour la science des données (*datascience*)
©F.Pennerath, Mineure « Data Science », Centrale Supélec

Pour fonctionner et effectuer des traitements, l'application *Kraken-Benchmark* requiert, outre le langage Python 3, l'installation des *packages* suivants (certains ont déjà étaient aperçus dans les sections précédentes et d'autres étant déjà inclus dans le langage Python) :

1. l'ensemble des fichiers de l'application cités à partir de maintenant sont disponibles dans les Annexes C

6.1 Modélisation

- Pour réaliser les calculs et l'implémentation des **métriques d'évaluation** : *Sklearn*², *Numpy*³, *Python-Levenshtein*⁴, *Kraken API*⁵, et des *packages* inclus dans Python comme *difflib*, *math*, *collections* ;
- Pour réaliser les **visualisations de données** : *Matplotlib*⁶, *Seaborn*⁷, *Pandas*⁸ ;
- Pour le **pré-traitement des données textuelles** (découpage des phrases en mots et suppressions des mots les plus fréquents) : *NLTK*⁹

À côté de ses *packages* orientés science des données, j'ai eu besoin de recourir à des *packages* plus spécifiques :

- Traitement des images en entrée : *Pillow*¹⁰ ;
- Réalisation de la chaîne de traitement HTR : *Kraken API* ;
- Gestion de la partie CLI : *argparse* ;
- Gestion de la partie GUI (*Graphical User Interface*) de l'application dans le navigateur *web* : *Flask*¹¹

Initialement au début du projet il était prévu de n'utiliser que le moteur de *templates*¹² *Jinja*¹³. Cependant, j'ai rapidement constaté les limites, notamment quand je devais permettre à l'utilisatrice ou à l'utilisateur de pouvoir accéder à plusieurs pages dans le navigateur (plusieurs vues), un gestionnaire de *templates* seul n'était pas suffisant.

L'avantage de *Flask* repose sur le fait qu'il s'agit d'une boîte à outils (mini *framework web*) pour construire des applications *web*. Ainsi en plus d'intégrer le moteur de *templates* *Jinja*, il intègre également un gestionnaire de routes (les différentes routes URL pour définir les vues), et dispose d'un gestionnaire de requêtes HTTP ainsi que d'un serveur de développement¹⁴ pour tester l'application¹⁵ et pour communiquer avec le code

2. *scikit-learn 0.22.1*, URL : https://scikit-learn.org/stable/user_guide.html

3. *numpy 1.19*, URL : <https://numpy.org/doc/stable/>

4. *python-Levenshtein 0.12.0...*

5. *Kraken API...*

6. *matplotlib 3.1.3*, URL : <https://matplotlib.org/contents.html#>

7. *seaborn 0.10.1*, URL : <https://seaborn.pydata.org/>

8. *pandas 1.0.3*, URL : <https://pandas.pydata.org/docs/>

9. *nltk 3.5...*

10. *pillow 7.2.0...*

11. *Flask 1.1.2*, URL : <https://flask.palletsprojects.com/en/1.1.x/>

12. Un moteur de *templates* permet de séparer la partie traitement pur des données en Python de la partie visuelle de l'application. Le *template* d'un projet *web* contiendra tous les fichiers HTML (pour la structure), le CSS et les images (style et aspect visuel du site), et le Javascript (pour les animations).

13. *Jinja 2.11*, URL : <https://jinja.palletsprojects.com/en/2.11.x/> (visité le 18/09/2020)

14. On distingue le serveur de développement, pour tester les applications *web*, du serveur de production, qui permet de déployer et d'installer l'application *web* finalisée qui est donc accessible aux utilisateurs.

15. **HTTP** (HyperText Transfer Protocol), est un protocole de communication entre client et serveur pour le *web*.

Python (ou WSGI¹⁶) *Werkzeug*¹⁷.

La structuration des éléments de texte (titres, description, paragraphes) de l'interface graphique a été réalisé à partir des *templates Jinja*, qui permettent de faire des appels d'objets Python directement dans le fichier HTML pour y insérer d'autres éléments comme les métriques. L'apparence générale du site (fenêtre principal, boutons, formulaires, bannière etc.) repose sur le *framework Bootstrap*¹⁸ qui est une collection d'outils de stylisation pour les sites *web* qui contient une feuille de style CSS déjà écrite. De plus, afin de rendre le site plus dynamique (symboles de chargement (*loader*), apparition des éléments au défilement etc.), j'ai ajouté des scripts en JQuery à la fin des pages HTML¹⁹.

La plupart des *packages* Python listés ci-dessus sont disponibles avec la distribution Python *Anaconda*. Il s'agit d'une distribution de Python comprenant le langage Python 3 mais également un ensemble de *packages* dédiés à la science des données. Ainsi plutôt que de devoir installer les *packages* un à un via le gestionnaire de base *pip*²⁰, l'utilisateur pourra créer un environnement virtuel *conda* par le biais du fichier `environment.yml` qui regroupe les *packages* cités plus haut.

L'installation et les commandes pour utiliser l'application sont résumés dans un fichier de type « lisez-moi » (`README.md`).

16. WSGI ou *Web Server Gateway Interface*, est interface entre des serveurs et des applications *web* pour Python. c'est une spécification qui permet à l'application Python de recevoir une requête HTTP transformée en un objet Python et de retourner une réponse HTTP sous la forme d'un autre objet Python également.

17. *Werkzeug*, URL : <https://werkzeug.palletsprojects.com/en/1.0.x/> (visité le 18/09/2020)

18. Mark Otto et Jacob Thornton, *Bootstrap*, version 4.5, URL : <https://getbootstrap.com/>

19. JQuery est une bibliothèque JavaScript pour faciliter l'écriture de scripts côté utilisatrices ou utilisateurs.

20. **pip** est le gestionnaire de *packages* Python par défaut. Il permet d'installer des *packages* supplémentaires dans la distribution Python classique par l'intermédiaire d'une commande de type `pip install nom-du-paquet`.

6.1 Modélisation

6.1.3 Les étapes de fonctionnement : retour sur quelques aspects de programmation

Après avoir rassemblé les outils pour réaliser le programme, j'ai modélisé les différentes étapes qu'il devait réaliser. De la même manière que pour développer l'outil *Generator Lectaurep-TEI* abordé dans la section 4.5, j'ai créé un modèle résumant l'algorithme de l'application (Cf. Annexes, Figure E.3).

Le programme se résume aux étapes suivantes :

1. Avant de lancer, l'utilisatrice ou l'utilisateur place dans différents dossiers les fichiers à traiter. À la racine du *script* principal `kraken_benchmark.py` l'utilisatrice ou l'utilisateur peut créer le dossier `dataset_GT` qui reçoit les fichiers en texte brut de la vérité terrain, le dossier `images` qui reçoit les images (format .jpeg) à reconnaître, et dans le dossier `model` un modèle (.mlmodel) pré-entraîné avec *Kraken*. L'utilisatrice ou l'utilisateur doit conserver le même ordre de fichiers dans les dossiers `dataset_GT` et `images` en utilisant un label dans le nommage de fichiers. Ainsi les images pourront être notées `image_1`, `image_2` ... et les transcriptions correspondantes `transcription_1`, `transcription_2` ... Sans cet étiquetage le programme risque d'associer une image avec la mauvaise transcription et la mauvaise vérité terrain.
2. Une fois l'étape précédente réalisée, on peut lancer le programme via la commande `$ python kraken_benchmark.py`. L'utilisatrice ou l'utilisateur peut également spécifier des options derrière sa commande comme `--label` s'il ou si elle souhaite renseigner des métadonnées sur ces images, `--verbosity` s'il ou si elle souhaite avoir un rapport complet durant chacune des étapes du programme, et une option `--clean_text` qui permet, en outre, d'enlever la ponctuation et les caractères non alphabétiques, ce qui peut être utile pour certains types de projets.
3. Le *script* principal `kraken_benchmark.py` lance une première étape de transcription HTR (qui s'appuie sur les fonctions des modules de l'API *Kraken*). Après avoir associés les images avec leur transcription vérité terrain, le programme effectue le processus suivant : il charge le modèle de transcription et les images, les images sont binarisées, il procède ensuite à la segmentation des images pour repérer les coordonnées des lignes de textes et effectue les prédictions qui sont normalisées dans un encodage UTF-8 ;
4. Le *script* principal `kraken_benchmark.py` rassemble alors, par triplet : l'image, la vérité terrain et la prédiction ;

5. Chacun des triplets est transformé en objet `SynSemTS` appartenant au module `SynSemTS.py` sur lesquels on peut récupérer les différentes métriques et visualisations associées ;
6. Le groupe d'objets est alors récupéré par le module `kb_report` qui prend le relais et se charge d'ouvrir le navigateur *web* automatiquement sur un serveur de développement par défaut ;
7. L'utilisatrice ou l'utilisateur peut alors circuler dans l'application et afficher les différentes pages *web*. Le `script routing.py` du dossier `kb_report` est chargé d'envoyer et de recevoir les requêtes URL pour afficher la page HTML correspondante durant la navigation de l'utilisatrice ou de l'utilisateur, jusqu'à la fin de la session.

Dans les sections qui suivent nous avons souhaité relater quelques questions de programmation rencontrées au cours du développement de l'application, plutôt que de simplement montrer le code Python brut. Le code source de l'application est consultable dans l'Annexe C, et une documentation interne au code explicite le rôle de chaque fonctions codés sous la forme de *docstrings*²¹.

6.1.3.1 Un jeu de données pour réaliser des tests fonctionnels

Durant le développement de l'application, je devais disposer de transcriptions de vérité terrain et d'un modèle de transcription pour tester l'application au fur et à mesure des ajouts de fonctionnalités. J'ai constitué un jeu de données en me basant sur des extraits de l'ouvrage *Voyage au centre de la Terre* de Jules Verne (dossier `jules_verne_set_test`). Toutefois, il ne s'agissait pas de réaliser un OCR de qualité, mais simplement de récupérer des données de tests à passer dans l'application.

Dans un premier temps, j'ai trouvé sur *Gallica* une série de 8 images hétérogènes. Parmi celles-ci, certaines contenaient du texte imprimé sur toute la page et d'autres des illustrations, de tailles variables, dans l'optique de tester ces différences.

La deuxième étape a été de constituer des vérités terrains en me basant sur l'environnement de transcription de *Kraken*²² (Cf. Figure 6.2).

Après avoir lancé la ligne de commande correspondante, le canevas de transcription apparaît sur une page HTML qui comprend, à gauche l'image et à droite les lignes à transcrire. Cela peut s'apparenter à une version minimalistre de la plate-forme *eScriptorium*.

21. Les *docstrings* sont des chaînes de texte situées à certains endroits du code pour commenter ou documenter. Elles visent à rendre le code source plus explicite pour les personnes qui souhaiteraient le réutiliser ou le maintenir. Elles peuvent suivre des conventions comme la PEP 257, le modèle *Sphinx*, ou le style *Google Python*.

22. Précisons qu'il s'agit d'un environnement « dépassé » qui n'est plus compatible avec la dernière version de *Kraken* mais qui reste suffisant pour générer du texte.

6.1 Modélisation

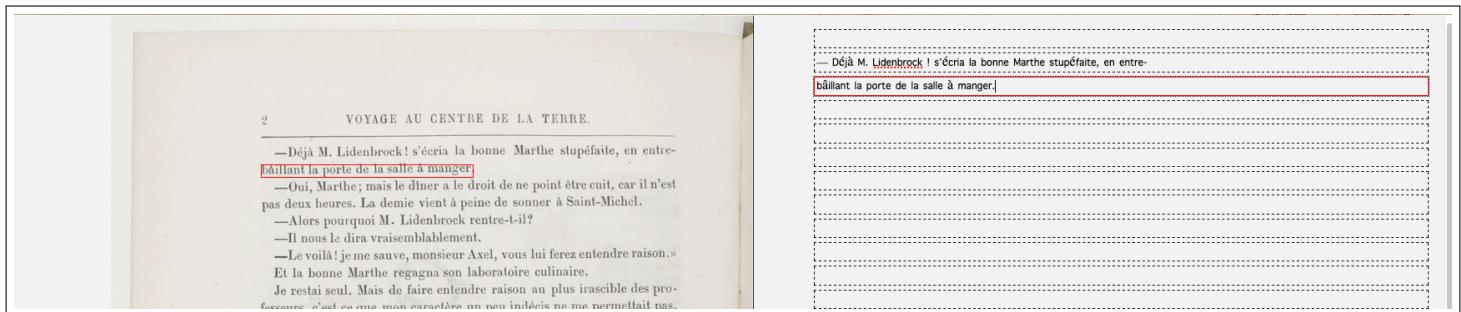


FIGURE 6.2 – Environnement de transcription de Kraken utilisés pour créer des vérités terrains de tests ©L. Terriel, 2020, *Kraken*

Une fois les transcriptions effectuées, une ligne de commande permet de récupérer les données d’entraînement avec *Kraken* sous la forme de fichier texte (le nommage du fichier étant de la forme `image_transcrite_1.gt.txt`) dans le dossier où l’on se situe. La dernière étape consistait par le biais d’une autre ligne de commande à entraîner un modèle de transcription à partir des vérités terrains constituées. Le modèle à pu être entraîné rapidement et les derniers résultats obtenus étaient plutôt satisfaisants (en cause les écritures imprimées lisibles).

6.1.3.2 Deux niveaux d’interface pour un prototypage rapide

La volonté de me concentrer rapidement sur la partie permettant d’exposer les métriques dans le navigateur et de disposer d’un outil, avant la fin du stage, pour effectuer des tests avec les données spécifiques de Lectaurep, m’ont obligé à faire un certain nombre de choix techniques.

Dans la mesure où il s’agit d’un prototype expérimental et interne à ALMAnaCH, il ne m’a pas paru nécessaire de doter l’application d’une interface graphique la partie chargement des données, traitement HTR/OCR, pré-traitements des données textuelles, et calculs.

J’ai séparé dès le départ :

- Une partie CLI, qui repose sur le *script* principal `kraken_benchmark.py` contenant la fonction chargée de réaliser la partie HTR. Ce *script* est relié à un module pour des traitements plus spécifiques `kraken_utils.py` : récupération du chemin des fichiers, impressions des messages de succès ou d’erreurs, construction de structures de données Python spécifiques, ou encore récupération des images dans le dossier `static` pour les afficher dans le navigateur. Le dossier `STS_Tools`, qui s’apparente à une « mini librairie » contient deux modules `SynSemTS.py` et `STSig.py` qui sont utilisés pour construire des objets sur lesquels on peut retrouver les métriques et les visualisations graphiques, dans le but de les afficher dans le navigateur (nous

reviendrons sur la spécificité de ces deux modules) ;

- Une partie GUI axée sur l'affichage dans le navigateur des métriques et des visualisations par l'ouverture du serveur de développement, le système de gestion des routes URL et des pages HTML par le module `routing.py` dans le dossier `kb_report` qui est appelé à l'issue du *script* principal `kraken_benchmark.py`.

Enfin, j'ai tenté de rendre l'affichage dans le terminal, pour la partie CLI, suffisamment convivial et explicite grâce à des *packages* Python. L'objectif étant que l'utilisatrice ou l'utilisateur puisse avoir des informations sur les étapes en cours. Chaque étape dispose d'une barre de progression qui indique le temps de traitement (*tqdm*²³), une colorisation indique bien à l'utilisatrice ou l'utilisateur les messages d'erreurs et les messages de succès (*termcolor*²⁴) et un affichage stylisé et une indication sonore alerte l'utilisatrice ou l'utilisateur que le programme est en attente d'informations (*prompt_toolkit*²⁵).

6.1.3.3 La programmation orientée objet : une solution pour généraliser et mieux documenter le code

Au fur et à mesure du développement de l'application la nécessité de rassembler les fonctions de calculs des principales métriques²⁶ en dehors du *script* principal `kraken_benchmark.py` s'est fait ressentir. En effet celui-ci se surchargeait de fonctions et la clarté du code devait plus difficile à maintenir.

Le recours à la programmation orientée objet (POO), rendue possible par Python, a été un bon moyen de contourner le problème et cela pour plusieurs raisons. Cependant, pourquoi ne pas avoir pris le parti de mettre les fonctions métriques dans un module comme `kraken_utils.py` qui regroupe des fonctions annexes au *script* principal ?

Les paragraphes suivants ont pour but de présenter brièvement la programmation objet, qui n'a pas la prétention de résumer un sujet aussi vaste, mais uniquement de cerner les enjeux, pour en venir à son utilisation concrète et à ses avantages pour l'application.

23. *tqdm* 4.46.1, URL : <https://tqdm.github.io/>

24. *termcolor* 1.1.0, URL : <https://pypi.org/project/termcolor/>

25. *prompt-toolkit* 3.0.5, URL : <https://python-prompt-toolkit.readthedocs.io/en/master/>

26. Décrives pour la plupart en section 5.2

6.1 Modélisation

La programmation objet est un paradigme de programmation qui consiste à structurer une application sur la base d'un assemblage d'entités indépendantes reliées entre-elles. C'est entités sont appelées « objets ». On peut voir un objet comme un concept du monde réel possédant des caractéristiques (en Python, on parle de propriétés), des comportements (méthodes), et peuvent avoir des interactions avec d'autres objets (Cf. Figure 6.3).

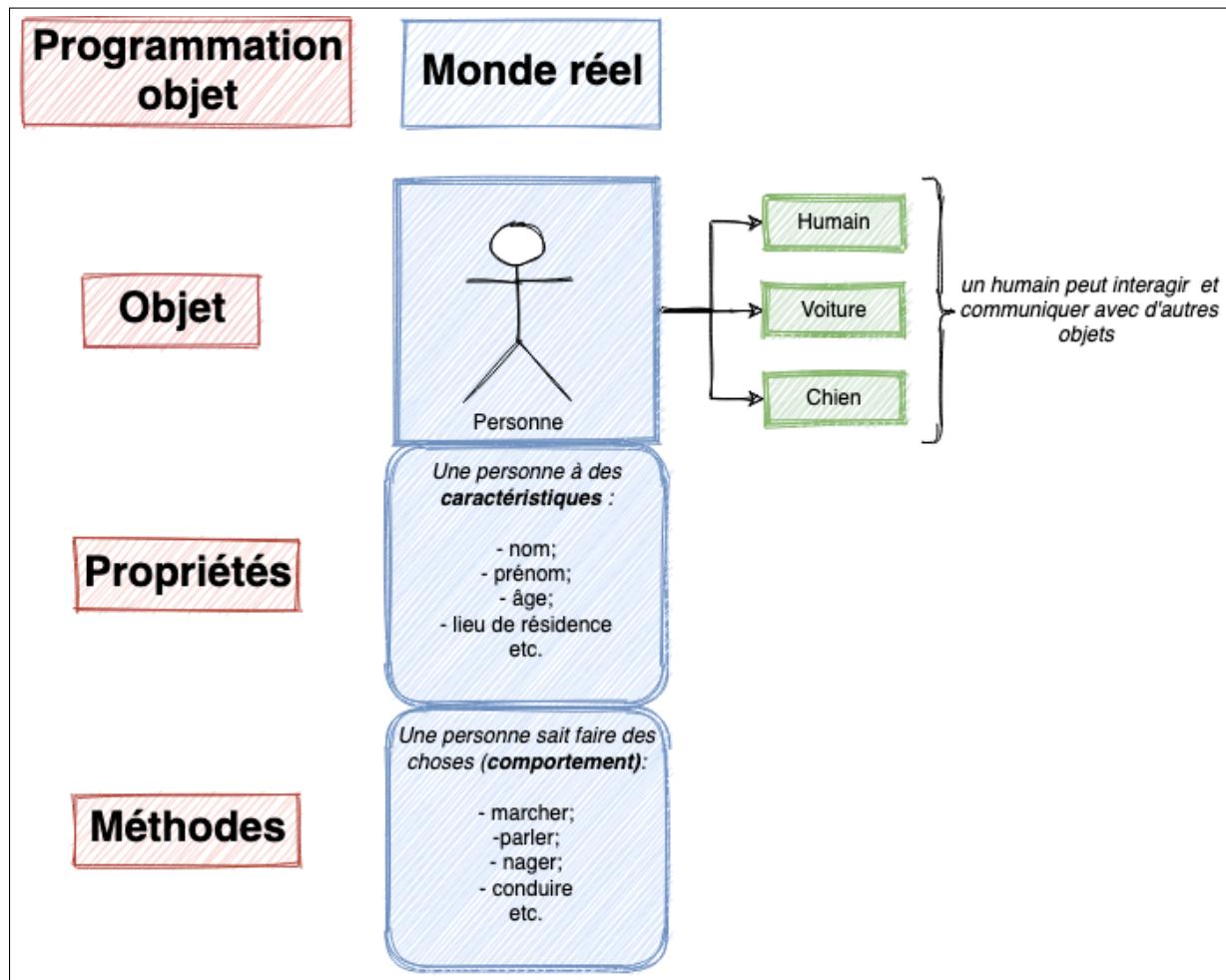


FIGURE 6.3 – Le concept de personne peut être représenté sous la forme d'un objet ©L. Terriel, 2020, Diagrams.net

Un objet peut relever d'une même catégorie. Ainsi une voiture est une sorte d'objet avec des caractéristiques et des méthodes propres pouvant appartenir à la catégorie moyen de transport. Celle-ci peut être modélisée au moyen d'une classe en langage orienté objet. La classe est apparentée à une usine qui va créer des objets héritant des propriétés de cette classe. L'exemple de code ci-dessous montre le moyen de créer un objet « Personne » en Python :

```

1 """On définit une classe Personne"""
2 class Personne:
3     def __init__(self, nom, prenom, age, lieu):
4         """Le constructeur permet de placer les propriétés de l'objet"""
5         self.nom = nom
6         self.prenom = prenom
7         self.age = age
8         self.lieu_de_naissance = lieu
9         """propriétés spécifiques pour les actions de l'objet"""
10        self.nombre_de_pas = 0
11    """On donne des comportements aux objets (méthodes)"""
12    def marcher_en_avant(self):
13        self.nombre_de_pas += 1
14        return print(f'{self.prenom} {self.nom} a fait {self.nombre_de_pas} pas !')
15    def parler(self, mot):
16        return print(f'mon mot est : \'{mot}\'')
17
18 """On crée un objet Personne (instanciation de l'objet dans la classe Personne)"""
19 Personne_1 = Personne("Mabillon", "Jean", 388, "Saint-Pierremont")
20
21 """On peut vérifier l'objet créé"""
22 print(Personne_1)
23 >>> <__main__.Personne object at 0x1013862d0>
24
25 """On peut récupérer des propriétés de l'objet"""
26 print(Personne_1.prenom)
27 >>> Jean
28
29 """On peut faire agir l'objet"""
30 Personne_1.marcher_en_avant()
31 >>> Jean Mabillon a fait 1 pas !
32 Personne_1.parler('De re diplomatica')
33 >>> mon mot est : "De re diplomatica"

```

6.1 Modélisation

C'est cette technique de programmation que j'ai mis en pratique afin de créer les deux modules `SynSemTS.py` et `STSig.py`. Nous aurons l'occasion de revenir sur ce dernier module d'expérimentation créé pour tester de nouvelles métriques et qui est encore en phase de développement, dans la section 6.2.2.

Le diagramme présenté en figure 6.4, ci-dessous, montre les différentes classes du module `SynSemTS.py`.

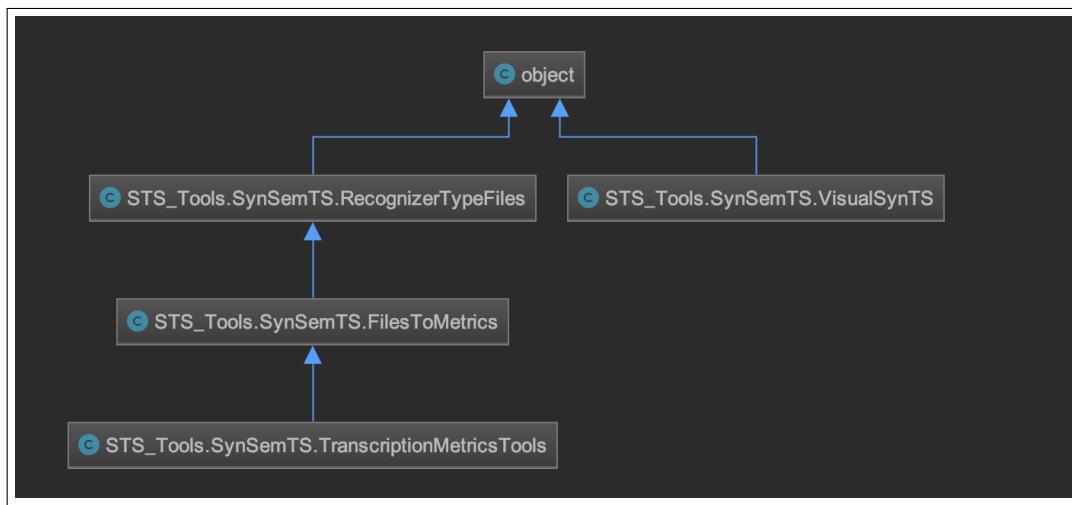


FIGURE 6.4 – Diagramme du module `SynSemTS.py` présentant les différentes classes ©L. Terriel, 2020, *Pycharm*

En reprenant le diagramme, une première classe (`RecognizerTypeFiles`) permet d'identifier les différents types de fichiers correspondant à la transcription vérité terrain, à la prédiction, à l'image et permet également d'effectuer certains traitements textuels sur les fichiers texte, comme le découpage en mots et en caractères.

Une seconde classe (`FilesToMetrics`) qui hérite des propriétés et méthodes de la première, effectue la transition entre la première classe et la troisième, en réalisant les calculs de la distance de Levenshtein et en récupérant les résultats sous forme d'entier et de matrice. Ce résultat permet en outre, d'effectuer les futurs calculs.

Une troisième classe (`TranscriptionMetricsTools`), la plus importante, hérite de la première et de la deuxième. Elle se charge des dernières tâches de normalisation textuelle (suppression des *stop words*) et réalise les calculs permettant de retrouver les métriques qui seront affichées plus tard dans le navigateur *web* (distance de Hamming, WER, CER, indice de Jaccard et similarité cosinus). Dès lors, les objets créés à la fin du *script* principal `kraken_benchmark.py` sont issus de la dernière classe.

Une autre classe est créée indépendamment (`VisualSynTS`) ; elle permet de réaliser des objets spécifiques correspondant aux visualisations graphiques affichées dans le navigateur *web*. En revanche, pour des raisons de temps de chargement trop longs, j'ai décidé de créer ces objets dans les routes (`routing.py`) au moment où l'utilisatrice ou

l'utilisateur formule une requête pour les consulter.

Pour conclure, il n'était pas obligatoire de programmer avec ce paradigme, mais les avantages de la programmation orientée objet sont nombreux et ont séduit pour la suite du projet.

Chaque objet présente une documentation claire quant à ses propriétés et ses méthodes, ce qui permet de savoir quels types de fonctions on manipule pendant le développement. Les types d'objets créés peuvent servir de base pour d'autres objets (on évite ainsi de réécrire du code existant). On peut réutiliser ces objets dans le cadre d'autres projets en réutilisant certaines briques déjà posées dans ces classes.

Enfin en plus d'obtenir un code plus compréhensible, il est plus facile de le maintenir et de le faire évoluer. On pourra modifier les objets, par exemple, ajouter de nouvelles métriques ou les corriger, sans toucher à l'ensemble de l'interface comme la séquence HTR ou la partie affichage dans le navigateur.

6.1.3.4 Retour sur les principales difficultés rencontrées

Parmi les difficultés auxquelles je me suis heurté au cours du développement de l'application, on peut relever les points suivants :

- La gestion des données d'entrée (fichiers texte, images et modèle) n'est pas encore optimale. Actuellement, l'utilisatrice ou l'utilisateur doit stocker ses fichiers à la racine du *script* principal suivant un système d'étiquetage comme on l'a vu plus haut. Ce système pourrait être amélioré ;
- Durant le développement de la fonction coïncidant avec la séquence HTR, je devais trouver le moyen de stocker et d'associer les fichiers entre-eux sans perdre l'ordre des fichiers pour ne pas mélanger une prédiction avec sa vérité terrain et son image. Je pouvais aboutir à des structures de données trop complexes (liste imbriquées de tuples) à manipuler. Je devais souvent lancer le programme en entier, plusieurs fois, pour être sûr que l'ordre n'avait pas été modifié. La programmation objet m'a permis de résoudre le problème ;
- N'étant pas familier de la programmation objet au moment de coder l'application, beaucoup d'essais ont étaient nécessaires pour aboutir au résultat actuel. Le module *SynSemTS.py* doit encore pouvoir être simplifié. Une réflexion doit être menée sur cet aspect qui permettrait d'accélérer davantage le processus de création des objets à la fin du script principal ;
- Je n'ai pas réussi à trouver le moyen d'afficher deux visualisations graphiques différentes sur la même route URL.

6.2 Suivi sur la conception et retour sur les usages de *Kraken-Benchmark*

6.2.1 La gestion du projet *Kraken-Benchmark*

Durant le développement de l'application, un ensemble de rituels a été mis en place afin de s'assurer des retours réguliers lors du développement de l'application.

Revue de code et intégration continue - Dans un premier temps, afin de vérifier que le code répondait aux exigences des règles d'usages propres au langage Python (obtenir un code « pythonique »), et qu'il était suffisamment compréhensible, un système d'intégration continue a été rendu accessible sur la plate-forme *GitLab* par Alix Chagué, au moment de mon arrivée²⁷. À chaque fois que j'ajoutais une fonctionnalité ou que j'effectuais une modification dans mon code, je le déposais sur *GitLab* par l'intermédiaire du système de versions *Git* : j'indiquais dans un message les modifications effectuées (commande *git commit*) et j'envoyais le code sur la plate-forme (commande *git push*). Au moment où la plate-forme Gitlab recevait le code, le système d'intégration continu exécutait automatiquement un fichier BASH (`ci-test.sh`²⁸), rédigé par Alix Chagué.

Ce fichier réalisait un test sur le code source, par le biais du logiciel *Pylint*, pour effectuer une vérification de qualité. *Pylint* utilise les recommandations officielles de la PEP8²⁹ pour s'assurer que les règles de rédaction du code sont respectées. Si le score était supérieur à 7, le système me renvoyait un message de succès. Ma responsable pouvait alors entreprendre une revue « manuelle » du code (*code review*) et se concentrer sur ses aspects fonctionnels du programme, une fois que j'avais effectué des propositions de modifications (*Merge Requests*). Dans le cas contraire, je devais reprendre mon code et repasser les tests.

Ce système, bien qu'en apparence contraignant, m'a obligé à assurer une rigueur tout au long de la rédaction de mon code, au fur et à mesure des développements et m'a permis d'accroître mes réflexes tels que la rédaction d'une documentation pour chaque fonctions, nommage des variables etc.

27. Dans le cadre de mon stage il s'agissait du logiciel *Jenkins* interfacé avec la plate-forme Gitlab.

28. Ce fichier est disponible dans les Annexes C

29. Les PEP (*Python Enhancement Proposal*), sont connues au sein de la communauté Python pour être des propositions d'améliorations du langage Python qui portent toutes un numéro. La PEP8 peut être vue comme un guide pour rédiger en langage Python et a pour objectif de définir des règles de développement communes entre développeurs : vérifier les indentations du code, les noms de variables, les espaces ou les lignes trop longues ; PEP8, URL : <https://www.python.org/dev/peps/pep-0008/>

Tests fonctionnels et retours utilisateurs - Sur ce point, les tests fonctionnels ont primés sur la rédaction de tests unitaires. On entend par tests unitaires un ensemble de scripts rédigés en langage Python, qui permettent de tester et de s'assurer du bon fonctionnement de plusieurs parties spécifiques du code. *Kraken-Benchmark* est considéré comme une application « non-critique » et qui fait une large part à l'expérimentation ; elle est actuellement utilisée par un nombre restreint de personnes au sein d'ALMAnaCH. Une fonctionnalité défaillante n'entraîne donc pas de retard conséquent pour l'avancement du projet, contrairement à une application comme *eScriptorium*, par exemple, qui doit être opérationnelle pour un grand nombres d'usages.

De plus, les difficultés rencontrées pour développer certaines fonctionnalités essentielles de l'application, conjuguées au besoin de pouvoir effectuer des tests sur des données Lectaurep avant la fin du stage, m'ont obligé à me concentrer sur l'aspect général et les résultats visibles de l'application, plutôt que de couvrir, par la rédaction de tests longs, chaque détails de fonctionnement de l'application.

Cependant, si l'outil doit évoluer dans la suite du projet, ou inclure de nouvelles fonctionnalités et hypothétiquement être mis en production à plus grande échelle, la rédaction de tests unitaires s'avérera être une priorité absolue afin s'assurer des bases plus solides.

Pour être sûr que l'application fonctionnait à chaque modification importante, je la relançait avec mon *set* de test (dossier `jules_verne_set_test`). Je pouvais ainsi m'assurer que le code fonctionnait toujours et qu'aucun conflit majeur n'était apparu.

Enfin j'ai créé une « zone de test », sous la forme d'un rapport d'erreurs (*issue*) sur *GitLab* pour recueillir les témoignages d'autres utilisateurs. Ainsi Florianne Chiffoleau, ingénierie en recherche et développement au laboratoire ALMAnaCH, a eu l'occasion de tester les performances de ses modèles de transcription dans *Kraken-Benchmark* afin de pouvoir les départager, dans le cadre du projet DAHN/MESRI traitant de la correspondance de Paul d'Estournelles de Constant (1852-1924).

Ces retours positifs montrent que l'utilisation de l'application peut être généralisée à d'autres projets d'HTR que Lectaurep.

6.2 Suivi sur la conception et retour sur les usages de Kraken-Benchmark

6.2.2 Un tour d'horizon de l'interface *Kraken-Benchmark* et des fonctionnalités actuelles

Nous proposons dans cette partie de passer en revue les fonctionnalités implémentées actuellement dans Lectaurep et la manière dont on peut interpréter les résultats.

La page d'accueil : tableau général de métriques et visualisation par images (Figure 6.5 et Figure 6.6) - La page d'accueil est la première page rencontrée par l'utilisatrice ou l'utilisateur. Un tableau de métriques (*Dashboard*) permet d'obtenir rapidement des scores sur l'ensemble de son corpus (Figure 6.5). Ces métriques ont été présentées en section 5.2 de ce mémoire. On peut attirer l'attention sur la distance de Hamming. C'est l'indicateur qui permet de révéler rapidement à l'utilisateur si la taille de la vérité terrain et de la prédiction font la même longueur. Dans le cas contraire, un symbole « \emptyset » l'indiquera. Un code couleur renseigne les meilleurs scores obtenus (vert) et les scores moins bons (rouge). De plus, sur la suggestion d'Alix Chagué, des hyperliens placés sur les numéros des images permettent d'accéder à l'image en question avec un niveau de détails plus important (Cf. Figure 6.6). À l'échelle de l'image, certaines métriques du tableau sont répétées afin d'éviter des alternances à l'utilisateur entre le haut et le bas de la page.

Un premier graphique montre le nombre de caractères exactement reconnus, supprimés et insérés. Enfin, il est possible d'accéder aux autres fonctionnalités qui montrent d'autres vues de l'application.

Overview								
Image	1	2	3	4	5	6	7	8
CER	0.07	0.07	0.09	0.11	0.12	0.18	0.41	0.18
CER (%)	7.48 %	7.09 %	9.83 %	11.69 %	12.57 %	18.45 %	41.6 %	18.12 %
WER	0.23	0.28	0.35	0.43	0.42	0.53	0.71	0.41
WER (%)	23.07 %	28.14 %	35.53 %	43.26 %	42.99 %	53.53 %	71.27 %	41.85 %
Word accuracy	76 %	71 %	64 %	56 %	57 %	46 %	28 %	58 %
Hamming distance	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Jaccard Index	0.51	0.4	0.31	0.27	0.3	0.28	0.23	0.31
Cosine Similarity	0.77	0.79	0.76	0.73	0.67	0.6	0.47	0.75

FIGURE 6.5 – La page d'accueil : tableau de métriques ©L.TERRIEL, 2020, *Kraken-Benchmark*

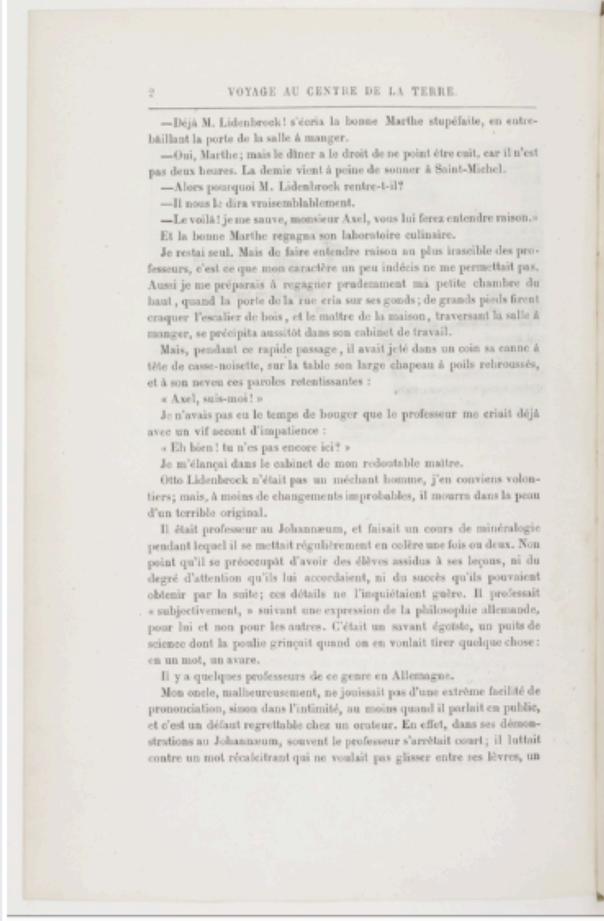
Report by Images

Image 1

Filename : Voyage_au_centre_de_la_[...]Verne_Jules_btv1b8600259v_16.jpeg

length reference : 2551

length prediction : 2464



Source gallica.bnf.fr / Bibliothèque nationale de France

Syntactic performance recognition metrics

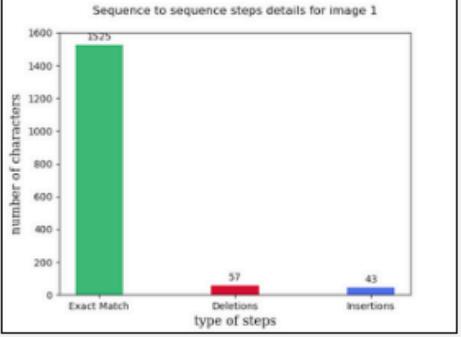
CER (Character Error Rate) : 7.48 %

WER (Word Error Rate) : 23.07 %

Word accuracy : 76 %

Click on image to enlarge

Sequence to sequence steps details for image 1



Type of steps	Number of characters
Exact Match	1525
Deletions	57
Insertions	43

Show versus

Ranking errors

Vizualize signals

Semantic performance recognition metrics

Jaccard Index : 0.51

Cosine Similarity : 0.77

FIGURE 6.6 – La page d'accueil : l'image et le graphique des opérations ©L.TERRIEL, 2020, *Kraken-Benchmark*

6.2 Suivi sur la conception et retour sur les usages de Kraken-Benchmark

Une option pour confronter la vérité terrain et la prédiction HTR (Figure 6.7) - La fonctionnalité *Show versus* de la page d'accueil permet de confronter la vérité terrain dans la colonne de gauche à la prédiction située dans la colonne de droite. La colonne du milieu est une superposition des deux textes. Un code couleur signale à l'utilisatrice ou à l'utilisateur les caractères parfaitement reconnus en vert, les caractères supprimés ou substitués en rouge, et les insertions en bleu.

Au niveau supérieur, la distance de Levenshtein indique à l'utilisateur la mesure des différences entre les deux textes : si le score est supérieur à 10, il s'affiche en rouge pour indiquer que le texte contient un grand nombre d'erreurs.



FIGURE 6.7 – La fonctionnalité *Show versus* ©L.TERRIEL, 2020, Kraken-Benchmark

Un classement des erreurs les plus fréquentes (Figure 6.8) - La fonctionnalité *Ranking errors* est un classement des erreurs les plus fréquentes commises par le modèle sur l'image. L'utilisatrice ou l'utilisateur dispose d'une indication sur la fréquence d'apparition de l'erreur et des détails sur le caractère de la vérité terrain qui a été confondue par le modèle. L'absence de caractère indique un « espace ». Sur la droite, l'utilisateur dispose d'une autre vue sous la forme d'une « matrice de confusion ». Il ne s'agit pas d'une « matrice de confusion » au sens strict, c'est-à-dire une classification des données par classes, il s'agit là du même classement des paires d'erreurs, uniquement pour les dix erreurs les plus fréquentes (sinon la matrice devient illisible).

Ainsi en ordonnée sont disposés les caractères du texte de référence et en abscisses les caractères de la phrase prédictive. L'alignement d'un caractère x et d'un caractère y nous donne la fréquence de la confusion. Le classement détaillé à gauche et la matrice à droite, sont équivalentes.

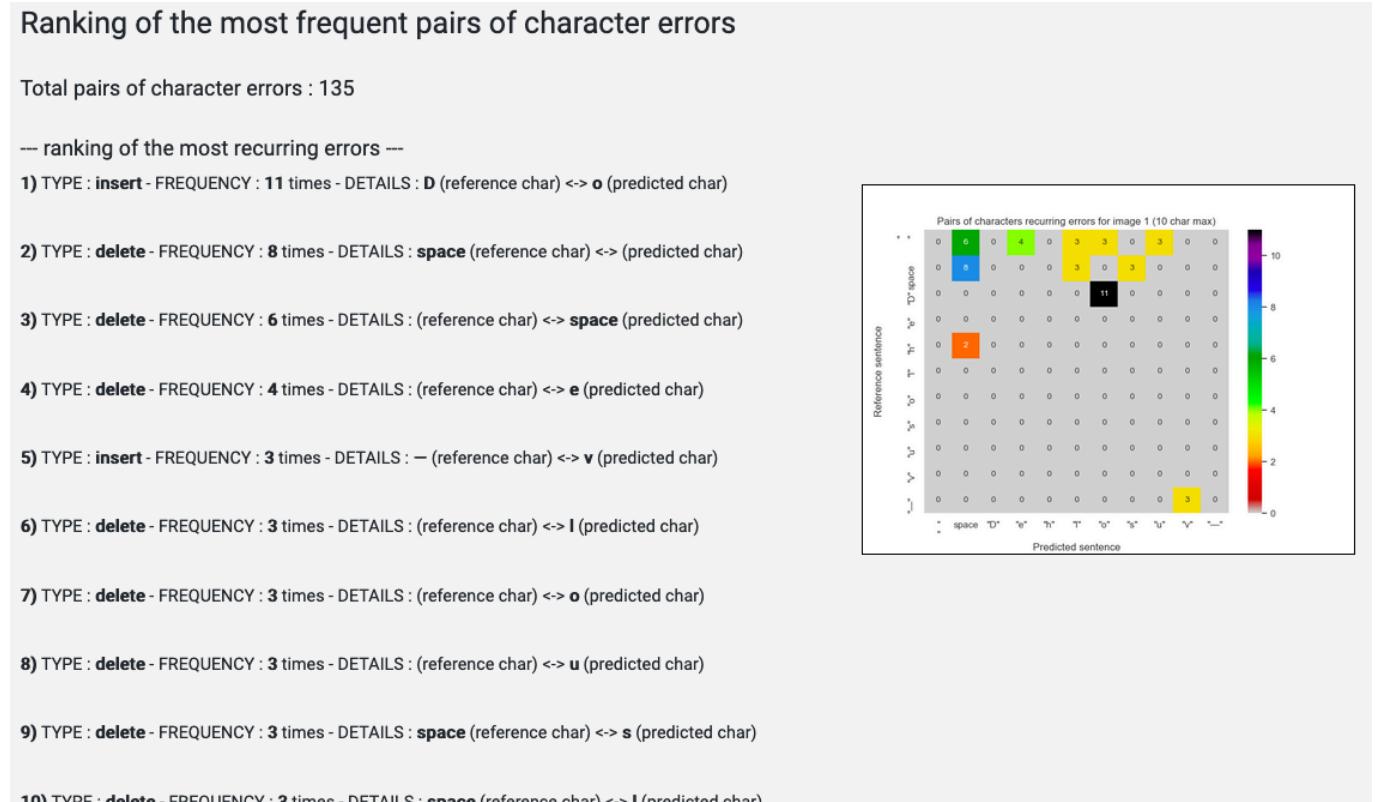


FIGURE 6.8 – La fonctionnalité *Ranking errors* ©L.TERRIEL, 2020, *Kraken-Benchmark*

Une visualisation des phrases sous la forme de signaux (Figures 6.9, 6.11 et 6.10) - La fonctionnalité *Vizualize signals* (Cf. Figure 6.9) permet de générer une visualisation expérimentale qui appartient au module *STSig.py*³⁰. J'ai essayé de superposer sous la forme de droites le texte de référence et la prédiction. Ces droites passent par des points dont les coordonnées correspondent, en abscisses, à l'ordre d'enchaînement des caractères dans les deux textes, et, en ordonnées, des valeurs numériques correspondant aux caractères, eux-mêmes accessibles par l'onglet *Dictionary position-weight characters*. Il s'agit d'un dictionnaire qui permet de décoder les valeurs numériques de l'axe des ordonnées.

30. Le détail de l'algorithme est présenté dans le *notebook* Jupyter intitulé « Evaluation de la similitude entre deux séquences dans le contexte de la reconnaissance automatique de caractères ».

6.2 Suivi sur la conception et retour sur les usages de Kraken-Benchmark

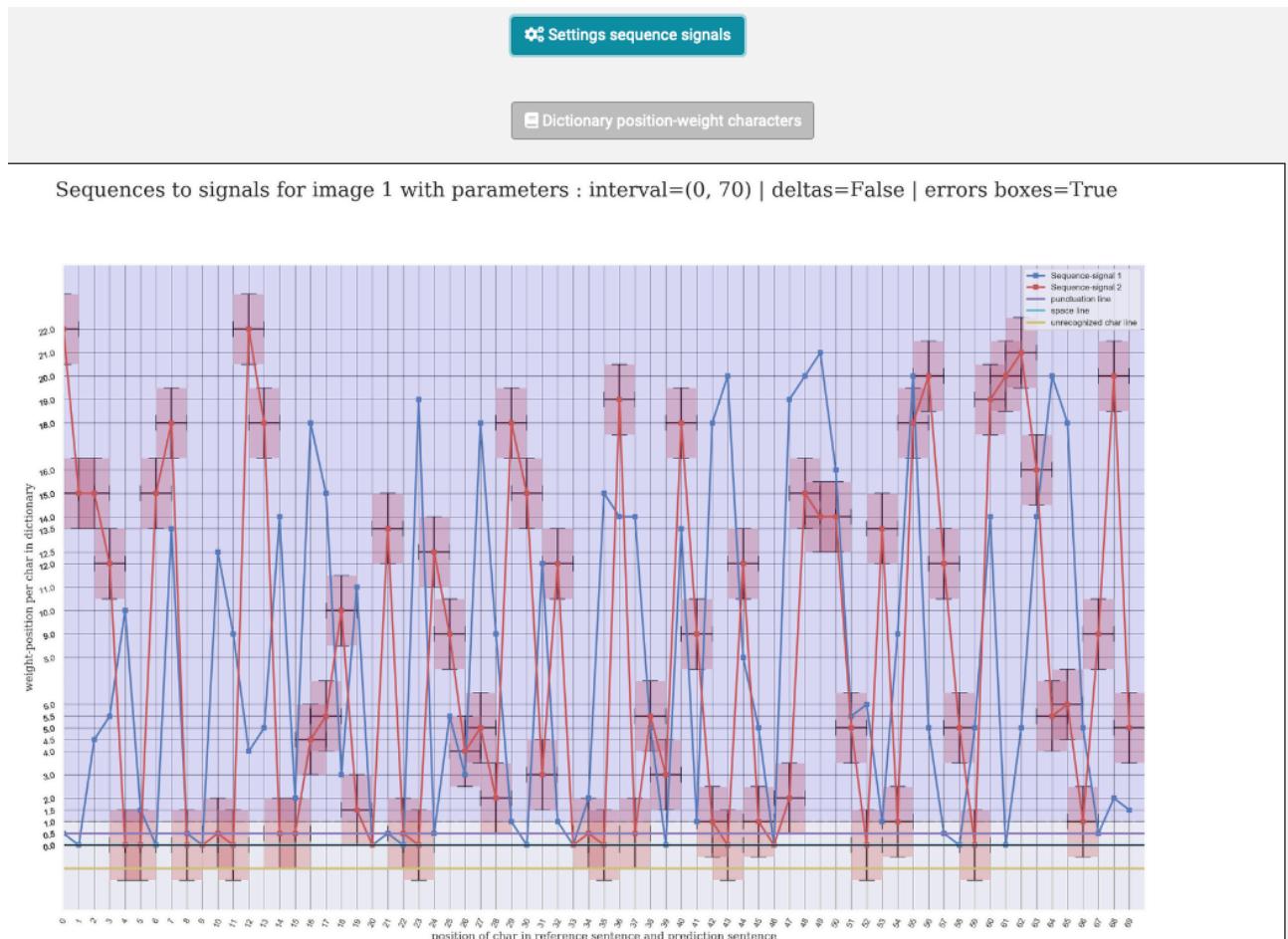


FIGURE 6.9 – La fonctionnalité *Vizualize signals* ©L.TERRIEL, 2020, *Kraken-Benchmark*

Ces valeurs numériques en ordonnées sont construites selon un système de pondération comme suit : un caractère de ponctuation sera associé à une pondération de 0.5, un espace à 0, et un caractère non reconnu ou des chiffres à une valeur de -1.

Chaque caractère de l'alphabet latin est encodé de la manière suivante : « a » vaudra 1, « b » vaudra 2, « c » vaudra 3 etc. Si ces caractères subissent une ou des transformation(s) on leur ajoute un poids : une accentuation vaudra 0.5, une capitalisation 0.1 et une capitalisation ajoutée à une accentuation 0.6. Par exemple si « a » vaut 1 alors « à » vaudra 1.5 et un « À » vaudra 1.6, de même si « b » vaut 2 alors « B » vaudra 2.1 etc. Dès lors, l'algorithme encode la vérité terrain et la prédiction selon ce système de valeurs numériques. L'algorithme récupère les coordonnées, pour représenter les deux textes sous la forme de droites. Si les deux droites se confondent, cela signifie que la prédiction correspond à la référence. Cependant si un caractère de la prédiction diffère de celui de la référence, une structure en boîtes d'erreurs (*error boxes*, représentées comme des boîtes rouges sur la figure 6.9) se met en place sur les points représentants les caractères de la prédiction divergeant de la référence. On peut alors interpréter certains phénomènes sur le graphique pour comprendre les erreurs récurrentes (Cf. Figure 6.10).

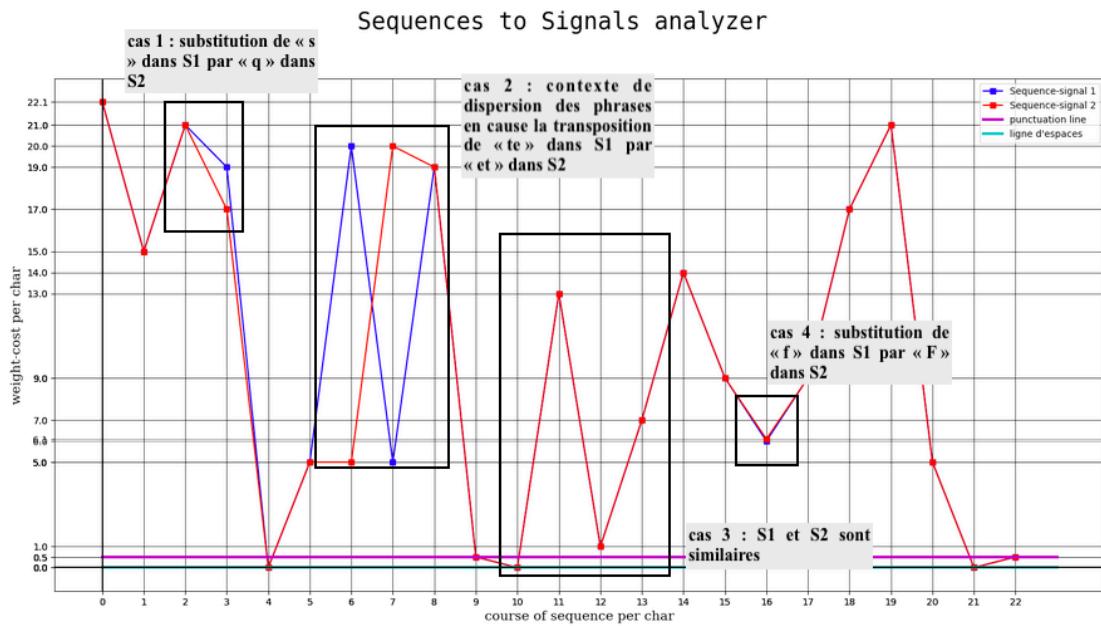


FIGURE 6.10 – Exemples d’interprétations de la fonctionnalité *Vizualize signals*
©L.TERRIEL, 2020, *Kraken-Benchmark*

Dans le but de ne pas surcharger le graphique et prendre le risque de le rendre illisible, un système d’intervalles est réglable (Cf. Figure 6.11) par l’utilisatrice ou l’utilisateur pour accéder à certaines parties des textes par le biais de l’onglet *Settings sequence signals*. Cette représentation est un essai, et elle ne peut se substituer aux métriques classiques d’évaluation des modèles HTR présentées plus haut ; de plus un trop grand nombre d’erreurs rend rapidement la lecture impossible.

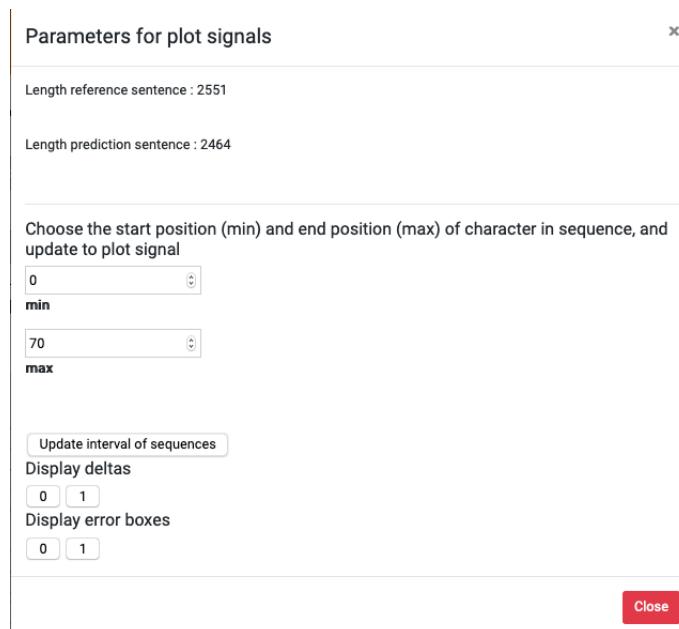


FIGURE 6.11 – La fenêtre « pop-up » qui permet de régler l’intervalle des séquences à visualiser dans le graphique de *Vizualize signals* ©L.TERRIEL, 2020, *Kraken-Benchmark*

6.3 Perspectives d'amélioration techniques pour l'application

6.3 Perspectives d'amélioration techniques pour l'application

Dans la section 6.1.1, nous avions fixé plusieurs objectifs à atteindre. À la fin du stage, certaines de ces tâches ont été réalisées et d'autres non pas pu l'être dans les temps.

L'outil proposé actuellement est opérationnel pour tester des modèles de transcription, regroupant les métriques classiques d'évaluation de l'HTR. Cependant, l'outil peut encore être largement amélioré, voici un potentiel cahier des charges de l'application :

- Dans la section 6.2.1, nous avons évoqué le manque de tests unitaires pour l'application ;
- La gestion des données en entrée peut bénéficier d'améliorations conséquentes : possibilité pour l'utilisatrice ou l'utilisateur de choisir un chemin sans avoir à trier ses données en amont du projet et de les étiqueter par un système de label dans le nommage des fichiers, le programme associant automatiquement la bonne image à sa transcription de vérité terrain. De plus, l'utilisatrice ou l'utilisateur doit pouvoir importer ses données dans des formats structurés comme le XML-TEI ;
- La seule manière de conserver le rapport est de sauvegarder la page HTML : l'application devrait proposer l'export du rapport dans des formats structurés comme XML-TEI ou XML ALTO ou encore en PDF. Le XML ALTO permet d'inclure des métriques comme le taux de confiance de reconnaissance (*word accuracy*) dans des éléments prédéfinis ;
- Un scénario envisageable pourra faire correspondre le fichier XML-TEI en import et en export dans l'application au format pivot XML-TEI de Lectaurep présenté dans la partie II (Cf. Figure 6.12)
- Certaines étapes de la séquence du traitement HTR dans l'application devraient être optionnelles (la binarisation par exemple) ;
- D'autres traitements TAL pourraient être proposés dans *Kraken-Benchmark*, ainsi l'application pourrait être interfacée avec l'API d'*Entity-Fishing* et paramétrée avec le référentiel souhaité pour tester les modèles spécifiquement entraînés à la reconnaissance d'entités nommées ;
- Actuellement, il n'est pas possible d'utiliser son propre modèle de segmentation dans l'application. À noter qu'au début du stage la documentation de l'API Kraken était moins fournie, ce qui n'est plus le cas ;
- Découlant de la proposition précédente, l'évaluation des modèles de segmentation dans l'application n'a pas été implémentée. Cependant la documentation de l'API Kraken enrichie depuis peu, ainsi qu'une rapide prospection des outils dans les

dépôts de code montrent que cela est réalisable³¹ ;

- La partie CLI, qui permet le chargement des données dans *Kraken-Benchmark*, constitue une solution provisoire. Elle pourrait bénéficier d'une interface graphique au même titre que la partie visualisation de données dans le navigateur *web*. Cela est possible en migrant le script principal `kraken_benchmark.py` vers un système de routes basé sur *Flask* et des *templates* HTML associées pour téléverser les fichiers dans une interface graphique ;
- Au lieu de se servir de l'actuel serveur de développement du *micro-framework* *Flask*, le programme pourrait reposer sur un serveur, comme Heroku³², pour permettre le déploiement de *Kraken-Benchmark* sur le *web* et sans avoir à lancer l'application depuis le terminal. Dans, une autre optique *Kraken-Benchmark* pourrait constituer une fonctionnalité (*plugin*) intégrée à *eScriptorium*³³. Ainsi l'utilisatrice ou l'utilisateur effectuant ses vérités terrains et l'entraînement de ses modèles sur *eScriptorium* pourrait jouir d'un relais vers *Kraken-Benchmark* pour mesurer l'efficacité de son modèle. Ce dernier point nécessite cependant d'avoir répondu à l'ensemble des tâches précédentes.

31. Le code du dépôt *TMG_ImageAnnotation* disponible sur GitHub propose un module d'annotation d'images qui pourrait être adapté pour les besoins d'évaluation de la segmentation pour *Kraken-Benchmark*. En combinaison avec les fonctions de l'API Kraken et le package Python *PIL* pour le traitement d'images ; on pourrait implémenter dans *Kraken-Benchmark* une visualisation de la segmentation sous la forme de rectangles entourant les zones de textes sur l'image indiquant si la segmentation a été correctement effectuée, via un code couleur. *TMG_ImageAnnotation*, URL : <https://github.com/guillotel-nothmann/imageAnnotation>

32. *Heroku - Cloud Application Platfrom*, URL : <https://www.heroku.com/>

33. *eScriptorium* est programmé avec le framework *Django* qui, comme *Flask*, repose sur une architecture de type MVT (Modèle-Vue-Templates) dès lors une migration vers ce projet doit être réalisable.

6.3 Perspectives d'amélioration techniques pour l'application

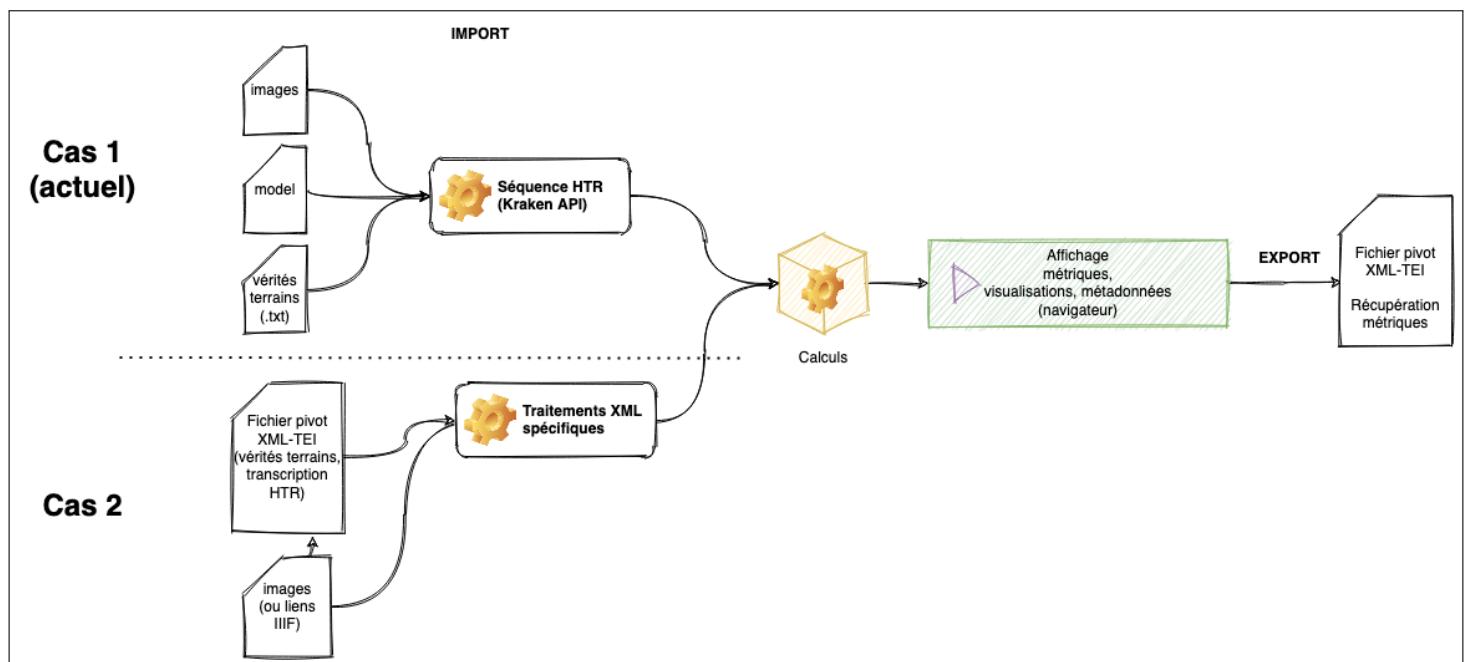


FIGURE 6.12 – Le Cas 2 représente un scénario utilisateur envisageable dans *Kraken-Benchmark* ©L. Terrie, 2020, *Diagrams.net*

Chapitre 7

Tests de *Kraken-Benchmark* sur les données Lectaurep

Au moment où *Kraken-Benchmark* a atteint une version « stable », j'ai envisagé d'expérimenter l'outil avec des données Lectaurep pour évaluer les modèles HTR et de simuler des usages répétitifs de l'outil en vue de la suite du projet¹.

7.1 Préparation des jeux de données

7.1.1 Les images

Durant le stage il a été convenu de tester l'outil sur des images « extrêmes », c'est-à-dire présentant des particularités d'écritures remarquables et des images de répertoire présentant des défauts physiques ou de numérisation. Aurélia Rostaing, responsable du pôle des instruments de recherche et coordinatrice du projet Lectaurep, ainsi que les annotatrices et les annotateurs du DMC sur la plate-forme d'*eScriptorium* ont effectué un travail de repérage important pour relever des images présentant des particularismes. A la réception du corpus d'images, j'ai constitué quatre jeux de données, composé chacun de trois images :

- *Set_material_defects* : jeu contenant des images présentant des défauts matériels ou des défauts de numérisation (Cf. Figure 7.1) ;
- *Set_writing_defects* : jeu composé d'images de répertoires comportant des éléments de graphies atypiques (typographies, signes et symboles) (Cf. Figure 7.2) ;

1. Pour davantage de détails, consulter le compte-rendu en Annexes C, `CR_tests_lectaurep_KB.md`. De plus l'ensemble des fichiers présentés dans cette section sont disponibles en Annexes C, `sets_tests_lectaurep/`



FIGURE 7.1 – *Set_material_defects* : les focus 1) et 2) correspondent à l'image `subject_1_robin_DAFANCH96_048MIC04695_L-1.jpeg`, (N&B, étude XLVIII, notaire Jean-François Robin), présentant des tâches d'encre et des écritures marginales, des noirceurs et des forts contrastes ; le focus 3) `subject_2_rigault_FRAN_0187_16416_L-1.jpeg`, (couleurs, étude LXXXVI, notaire Jean-Paul Rigault), présentant un ruban adhésif sur la partie inférieure droite du document obstruant une partie des colonnes 6 et 7 ; le focus 4) `subject_3_michaux_DAFANCH96_MIC067000672-1.jpeg`, (N&B, étude VII, Pierre Michaux) présentant un ruban adhésif épais obstruant une partie du texte dans les colonnes 1, 2, 3, et 4 et une très mauvaise qualité de numérisation. ©AN-DMC, 2020

7.1 Préparation des jeux de données

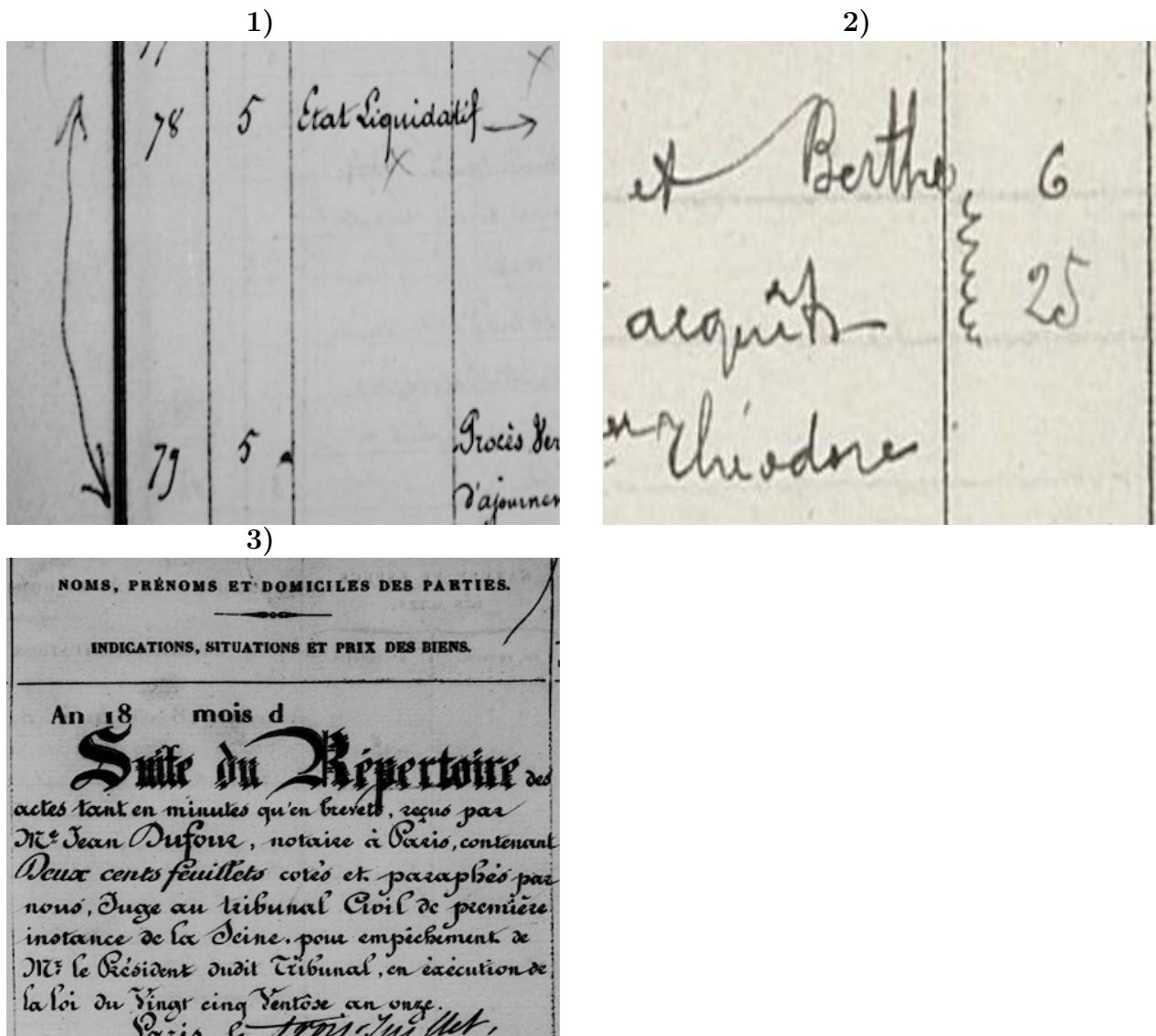


FIGURE 7.2 – *Set_writing_defects* : le focus 1) correspond à l'image subject_1_dufour_DAFANCH96_048MIC07685_L-0.jpeg, (N&B, étude XLVIII, notaire Jean Dufour), présentant une double flèche dans le coin inférieur gauche du répertoire au niveau de la colonne 1; le focus 2) correspond à l'image subject_2_rigault_FRAN_0187_16428_L-1.jpeg, (N&B, étude LXXXVI, notaire Jean-Paul Rigault), présentant un élément atypique situé au niveau du nombre « 25 » de la colonne 6 correspondant aux « dates »; le focus 3) correspond à l'image subject_3_dufour_DAFANCH96_048MIC08733_L-1.jpeg, (N&B, étude XLVIII, notaire Jean Dufour) présentant une calligraphie fracturée. ©AN/DMC, 2020

Viennent ensuite deux *sets* témoins pour compléter l’expérience et comparer les résultats :

- *homogeneous_control_set* : comportant des images n’ayant subit aucune interférence lors de la numérisation du corpus et présentant des graphies régulières, une présentation des écritures réparties dans les colonnes du tableau de manière homogène, sans altération matérielle apparente des répertoires de notaire, provenant de la même étude XLIII du notaire Louis Marie Joseph Marotte ;
- *different_control_set* : similaire aux caractéristiques du premier jeu de données présenté ci-dessus, à la différence près que les images proviennent chacune d’une autre étude.

7.1.2 Les modèles

Les modèles utilisés pour le test ont été entraînés par Alix Chagué et Floriane Chiffoleau à ALmanaCH, par le système HTR :

- `model_test_lectaurep_bin_accuracy_6064.mlmodel` : a été entraîné avec 135 pages du notaire Rigault et sur 23 *epochs*. Il s’agit du modèle de l’époque 18 avec un *accuracy* de 0.6064.²
- `model_test_lectaurep_bin_accuracy_8164.mlmodel` : a été entraîné au début du projet Lectaurep avec 67 pages du notaire Marotte et sur 14 *epochs*. Il s’agit du modèle de l’époque 9 avec un *accuracy* de 0.8164.

7.1.3 Les vérités terrains

Pour les deux *sets* présentant des particularismes, les vérités terrains ont été réalisées par Danis Habib, chargé d’études documentaires au DMC, sur la plate-forme *eScriptorium*. Elles m’ont été transmises dans un format texte.

Pour les *sets* témoins, les vérités terrains étaient disponibles sur la plate-forme *Sharedocs* au format XML ALTO. J’ai donc récupéré un script Python de la Bibliothèque d’État de Berlin³ qui permet la conversion du XML ALTO vers un format texte pour retrouver les vérités terrains compatibles avec *Kraken-Benchmark*.

2. Lectaurep disposait encore de trop peu de vérités terrains, de plus il y a eu des problèmes de compatibilité entre le serveur RIOC (*cluster* de calcul) et les dépendances de Kraken.

3. Ce script est disponible dans les Annexes C, `alto2text.py`

7.2 Méthodologie et résultats obtenus

Pour effectuer les tests j'ai passé successivement chacun des jeux de données, présentés plus haut, contenant les documents de vérité terrains, les images avec les deux modèles dans l'outil *Kraken-Benchmark*. J'ai établi un compte rendu (`CR_tests_lectaurep_KB.md`) dans lequel j'ai relevé l'ensemble des métriques obtenues et effectué des sauvegardes de la page HTML de la fonctionnalité *versus-text* de l'application qui permet la confrontation de la vérité terrain et de la prédiction⁴.

J'ai ensuite collecté spécifiquement les résultats WER, CER et le taux de reconnaissance par mot dans deux tableurs CSV⁵ (*Comma Separated Values*) et réaliser des moyennes globales pour chaque jeu de données à partir de ces taux. Ils sont reportés dans les deux tableaux ci-dessous :

Résultats modèle <i>lectaurep_bin_accuracy_6064</i>			
	Moyenne CER	Moyenne WER	Moyenne Word Accuracy*
homogeneous_control_set	76.78 %	94.64 %	5 %
different_control_set	72.30 %	93.0 %	7 %
Set_writing_defects	74.20 %	94.50 %	4 %
Set_material_defects	75.36 %	96.15 %	3 %

Résultats modèle <i>lectaurep_bin_accuracy_8164</i>			
	Moyenne CER	Moyenne WER	Moyenne Word Accuracy*
homogeneous_control_set	70.0 %	96.16 %	4 %
different_control_set	69.21 %	93.62 %	6 %
Set_writing_defects	73.0 %	96.70 %	3 %
Set_material_defects	73.0 %	97.46 %	2 %

* mesure arrondie à l'entier

Les résultats de ces deux séries de tests sont globalement mauvais quant aux perspectives de récupération d'une transcription propre et lisible avec ces modèles.

Si l'on observe les deux tableaux ci-dessus les moyennes du WER et du CER sont élevés : généralement entre 70% et 100%. Le taux de reconnaissance par mots (*word accuracy*) lui ne dépasse pas la barre des 10%. Le modèle commet des fautes sur l'ensemble des mots du texte. De plus, on remarque que les résultats sont à la fois très faibles et en même temps très proches dans le cas des deux modèles.

4. Ces captures sont disponibles dans les Annexes C, `snaps_tests_lectaurep/`

5. Ces tableurs sont disponibles dans les Annexes C, `details_data_average_tests_model_test_lectaurep_bin_accuracy_6064.mlmodel- Feuille 1.pdf` et `details_data_average_tests_model_test_lectaurep_bin_accuracy_8164.mlmodel- Feuille 1.pdf`

Durant les tests, certaines visualisations qu'offre *Kraken-Benchmark* n'ont pas pu être utilisées (classement des paires d'erreurs, et visualisations en signaux). En cause, le nombre important d'insertions et de suppressions élevés dans la prédiction (comme en témoigne les absences de distance de Hamming et les distances d'édition importantes). Le bruit trop intense dans la prédiction provoque des décalages trop importants entre les séquences de caractères correspondant à la référence et à la prédiction. Seul le *dashboard* de métriques et la confrontation à l'oeil nu de la référence et de la prédiction avec l'option *versus text* ont pu être utilisés.

7.3 Un bilan mitigé pour les tests dans *Kraken-Benchmark* ?

Si l'on regarde les taux obtenus par les modèles HTR les plus performants dans d'autres projets, ils restent inférieurs à 20% et peuvent être considérés comme lisible pour l'oeil humain⁶, en revanche des taux supérieurs à 70%, comme dans le cas des tests effectués ici, sont non seulement illisibles à l'oeil nu mais aussi moins précis pour un traitement informatique.

De plus comme l'avait noté Marie-Laurence Bonhomme lors de la phase préliminaire du projet : « On considère aujourd'hui comme bon un modèle d'HTR qui produit des résultats où le CER est inférieur à 10%, et comme très bons ceux dont le CER est autour de 5% ». ⁷. Dès lors, les modèles de transcription utilisés sont bien en deça des attentes de Lectaurep. Plusieurs pistes d'explication sont possibles :

6. Killian Barrere, Bertrand Coüasnon et Aurélie Lemaitre, *Results of a PyTorch implementation of an Handwritten Text Recognition Framework*, rapport, INTUIDOC Research group, 2018, URL : http://perso.eleves.ens-rennes.fr/people/killian.barrere/papers/Results_of_a_PyTorch_implementation_of_an_Handwritten_Text_Recognition_Framework.pdf ; Ioannis Pratikakis, Kostantinos Zagori, Panagiotis Kaddas et Basilis Gatos, « ICFHR 2018 Competition on Handwritten Document Image Binarization (H-DIBCO 2018) », dans *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, URL : https://eee.hbut.edu.cn/_local/9/3C/5B/599ED504CDE84710DC7A07A7ABF_A91E00B3_5F797.pdf?e=.pdf ; Chris Olver, *Machine learning of an 18th century hand : transcribing the essays of George III*, Georgian Papers Programme, 2017, URL : <https://georgianpapers.com/2017/01/20/machine-learning-18th-century-hand-transcribing-essays-george-iii/> (visité le 21/09/2020) ; Sofia Ares Oliveira et Frederic Kaplan, *Comparing human and machine performances in transcribing 18th century handwritten Venetian script*, DH2018, 2018, URL : <https://dh2018.adho.org/en/comparing-human-and-machine-performances-in-transcribing-18th-century-handwritten-venetian-script/> (visité le 21/09/2020) et Victor Lavrenko, Toni M. Rath et R. Manmatha, « Holistic word recognition for handwritten historical documents », dans *First International Workshop on Document Image Analysis for Libraries*, 2004, URL : <http://ciir.cs.umass.edu/pubfiles/mm-319.pdf>

7. M.L. Bonhomme, *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris...*, pp.7

7.3 Un bilan mitigé pour les tests dans *Kraken-Benchmark* ?

- Lectaurep n'est pas encore entré dans sa phase de transcription ; nous ne disposions pour ces tests que de très peu de données d'entraînements pour de la reconnaissance d'écriture, comme le modèle sous-ajusté (*underfitted model*) `model_test_lectaurep_bin_accuracy_6064.mlmodel` entraîné avec 135 pages. De plus, lors des entraînements de ce modèle, il existait de nombreux conflits entre les dépendances de la version de *Kraken* et l'utilisation du serveur INRIA RIOC.
- *Kraken-Benchmark* n'est pas encore capable d'intégrer des modèles de segmentation entraînés spécifiquement pour Lectaurep. L'application utilise actuellement une segmentation par défaut de *Kraken*, ce qui peut, en partie, tromper les résultats.
- Les modèles utilisés sont centrés généralement sur des notaires précis (Marotte et Rigault), ce qui peut constituer un biais d'entraînement et poser des problèmes lorsque le modèle doit rencontrer d'autres types de mains dans les répertoires.

Depuis la fin du stage, ma responsable a pu informer l'ensemble des acteurs de Lectaurep de l'entraînement de nouveaux modèles avec des données plus nombreuses : un modèle entraîné sur 18 688 lignes, soit 53 pages du notaire Marotte, avec un taux d'exactitude à 87,46% (proche des taux obtenu avec *Transkribus* lors de la phase 1), et un modèle entraîné sur 18 688 lignes, soit 100 pages du notaire Riant, avec un taux d'exactitude de 74,67 %, et un modèle du notaire Dufour entraîné sur 14 590 lignes, soit 89 pages du notaire Dufour, avec un taux d'exactitude de 76,46%.

Le taux d'exactitude correspond à un taux de performance du modèles produit durant les itérations d'entraînement (on parle de *Best Accuracy* pour décrire le modèle ayant obtenu le meilleur résultat. Le *Last Accuracy* (terme non conventionnel), décrit en Figure 7.3, correspond au taux d'exactitude du modèle de la dernière itération). Les résultats sont disponibles en Figure 7.3. Alix Chagué a pu observer que « le modèle qui obtient le plus haut taux d'erreur est aussi celui qui a produit le plus grand nombre d'itérations sur le corpus de vérité terrain le plus petit », une hypothèse qui reste à vérifier avec des données plus hétérogènes et plus nombreuses. De plus, ces modèles devraient être testés dans *Kraken-Benchmark* lorsque les modèles de segmentation propres à Lectaurep pourront s'interfacer avec l'application.

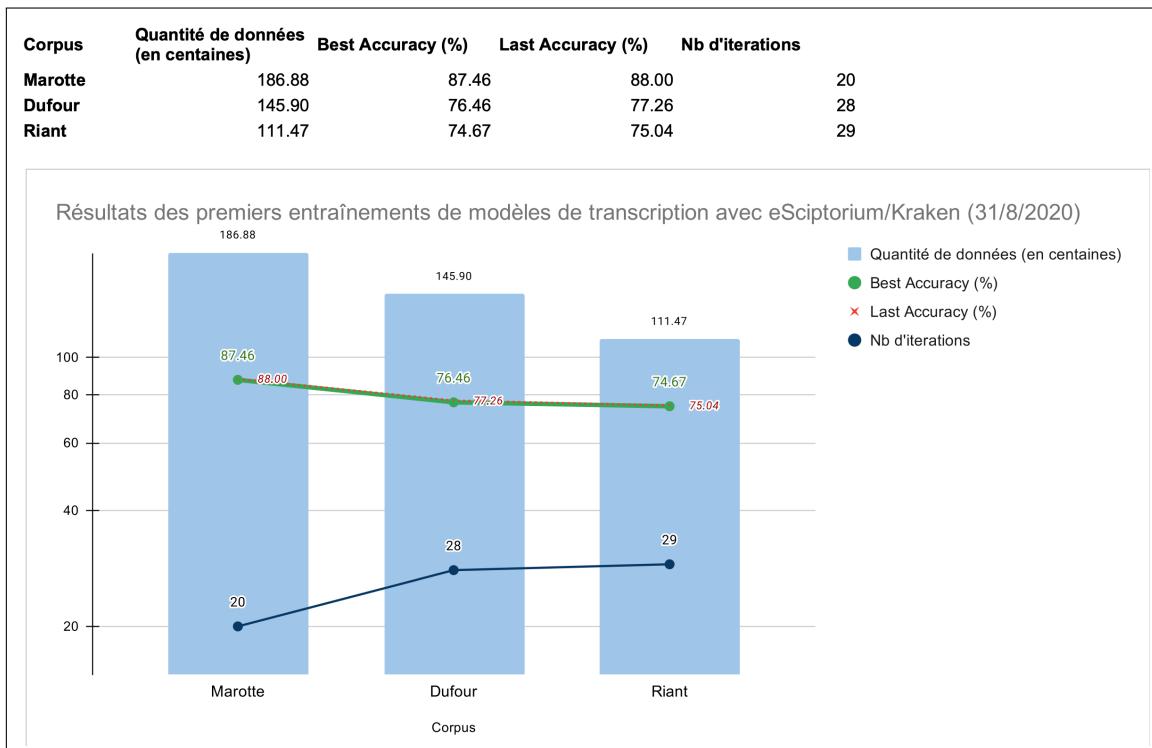


FIGURE 7.3 – Derniers entraînements de modèles réalisés pour Lectaurep ©A. Chagué, 2020

Lectaurep gagnerait à entraîner un modèle sur des données plus nombreuses et hétérogènes. Sur ce dernier point, le *Random set* sur l'espace *ShareDocs*, initialement créé pour entraîner un modèle général de données mixtes, contient sept cents pages de notaires très différentes, qui pourrait être utilisé pour créer un modèle capable de s'appliquer à une plus grande diversité de mains, quitte à l'affiner pour des données plus spécifiques par la suite (*Fine-tune*).

Cependant, l'ensemble des images de ce *Random set* n'ont pas été encore étaient transcrites, et sont en cours d'annotation. La stratégie actuelle du DMC repose sur la transcription des images apparaissant dans l'ordre du *Golden Set* - des lots de cent images homogènes (correspondant à un notaire)- afin de faciliter le travail des annotatrices et des annotateurs. Une approche compréhensible du point de vue du travail de transcription manuel, souvent fastidieux pour les annotatrices et les annotateurs, qui une fois avoir transcrit une ou deux pages de répertoire d'un notaire s'habituent rapidement aux écritures des autres images du même notaire ; en effet, il peut être difficile pour une annotatrice ou un annotateur de sauter d'une écriture à l'autre. On remarque, que la réalité du travail d'annotation peut entrer en conflit avec la création d'un modèle de transcription général qui doit disposer de données variées pour traiter des cas nouveaux. On peut prévoir que cette méthodologie évoluera au sein du DMC dans la suite du projet afin d'obtenir de meilleurs résultats.

Conclusion

Conclusion

Arrivé au terme de ce mémoire, quel bilan puis-je tirer, en terme de limites et d'avantages quant aux outils et réflexions que j'ai produit tout au long de ce stage ?

La présentation de la chaîne de traitement Lectaurep a révélé des structures de données complexes et difficilement interopérables entre-elles en l'état. De plus nous avons souligné la limite des outils comme *eScriptorium* qui nécessitent encore beaucoup de fonctionnalités et de maintenance.

Le modèle commun respectant le standard de la TEI que nous avons mis en place durant le stage est une solution à l'interopérabilité et à la récupération des données, comme les images, au sein d'*eScriptorium*. En posant un cadre de développement à travers une ODD et un script *Generator Lectaurep-TEI*, nous laissons un espace de travail ouvert afin d'affiner les futures réflexions sur l'architecture du canevas XML-TEI. En effet, ce format doit encore accueillir de nouvelles données comme les entités nommées, la structure logique des tableaux des répertoires de notaires et liens IIIF vers les images. Toutes ces informations restent à encoder et feront nécessairement évoluer la structuration XML-TEI proposée dans ce mémoire.

Des essais d'intégration dans la plate-forme *eScriptorium* et des projet d'éditorialisation des répertoires à partir du fichier pivot XML-TEI, restent à prévoir et permettront d'avoir des retours plus précis concernant les besoins spécifiques des utilisatrices et des utilisateurs d'*eScriptorium* en terme de récupération de données et de métadonnées.

L'outil *Kraken-benchmark* est actuellement utilisable pour effectuer les futurs tests des modèles de transcription. Cependant, si l'outil doit encore évoluer et passer à des usages à grande échelle, des optimisations techniques devront être réalisées. En priorité, une revue du code-source, la mise en place de tests unitaires, et l'intégration d'une fonctionnalité permettant d'intégrer des modèles de segmentation de Lectaurep. Ce sont là les conditions pour un éventuel portage de l'application sur un serveur de production ou comme une extension (*add-on*) d'*eScriptorium*.

Si les tests spécifiques effectués pour les données de Lectaurep ont pu être réalisés avec *Kraken-benchmark*, cependant la durée du stage ne m'a pas permis d'intégrer les résultats dans le fichier pivot XML-TEI. De plus, les mauvais résultats, dû à des modèles défaillants et à la non prise en compte des modèles de segmentation propres à Lectaurep, doivent permettre de poser un axe de réflexion quant aux stratégies actuelles d'entraînement des modèles par Lectaurep.

Pour la suite du projet, et en s'appuyant sur les principes élémentaires du *Deep learning* que nous nous sommes efforcés de résumer dans ce mémoire, les futurs modèles devront disposer de données : plus nombreuses - on estime qu'un modèle de segmentation efficace doit compter pas de moins trois cents à quatre cents pages et plus de cent itérations - et plus hétérogènes afin de travailler sur des jeux de données mixtes. Enfin, une réflexion doit également être menée, par le département du minutier central des archives nationales, en vue de formaliser des règles de transcription pour les annotateurs et les annotatrices d'*eScriptorium* qui préparent les vérités terrains, afin que ces données conservent une cohérence lors des entraînements.

Mon stage est une illustration du dialogue actuel des institutions patrimoniales avec les nouvelles technologies, entre les archivistes du DMC et les ingénieur(e)s d'ALMA-naCH. Des réunions mensuelles permettaient d'illustrer les avancés de mes recherches sur les missions que l'on m'avait confiées. De plus le travail à distance à pu être raccourci par les canaux de communication mis en place (*mattermost, zoom*). Grâce à ces outils Alix Chagué, notre responsable de stage, Jean-Damien Généro, stagiaire du master TNAH sur le projet *Time us* et moi-même pouvions collaborer, lors de deux réunions hebdomadaires sur les avancées et les tâches à réaliser pour la semaine suivante, mais également élargir nos horizons en faisant communiquer les approches des deux projets.

J'ai beaucoup appris durant ce stage et complété ma formation initiale. J'ai pu appréhender de nouveaux concepts en programmation (la programmation orientée objet, l'exploitation des structures de données plus avancées entre autres) et parfaire ma syntaxe en Python par le biais des conseils dispensés lors des revues de codes et du tutoriel présenté par ma responsable de stage. Mais également concernant l'aspect interopérabilité des données et les multiples usages de la TEI, présentés par Laurent Romary.

À compter du 1^{er} novembre 2020, je poursuivrai les missions débutées durant mon stage à ALMA-naCH. Notamment sur les prochaines étapes de Lectaurep qui consisteront à charger des images à partir de IIIF, mettre ces images à disposition des annotateurs et des annotatrices, et récupérer les métadonnées de traitements dans la SIV des Archives nationales. De plus je rejoindrai, le projet « NER4Archives », projet en partenariat avec les Archives nationales, visant à améliorer la description des outils de recherche encodés en EAD, grâce à la reconnaissance des différentes entités identifiables dans les champs correspondants. Ainsi que le projet européen « EHRI III » (*European Holocaust Research Infrastructure*), où l'objectif est d'intégrer des descriptions archivistiques issues du réseau international et d'unifier les représentations et d'en enrichir les contenus (vocabulaire, prosopographie).

Conclusion

Durant ce stage j'ai compris que l'ambition d'un projet alliant le patrimoine et le numérique était de rompre avec les pratiques traditionnelles de la recherche d'informations à destination des publics qui, pour la plupart maîtrisent les technologies numériques. Mais il ne faut pas oublier de prendre en compte l'écart qui peut encore être présent au sein de ce public : l'INSEE dans une récente étude, a révélé que l'illectronisme touche encore 17% de la population française ; la fracture numérique existe toujours⁸. Ces outils, basés sur l'intelligence artificielle, ne pourraient se suffire à eux-mêmes et ne seront réellement « intelligents » que s'ils sont utilisés. Le défi des institutions patrimoniales pour les années à venir sera d'accompagner les utilisateurs et les utilisatrices vers ces outils et les chercheuses et les chercheurs devront faire de l'informatique, non plus une science auxiliaire de l'histoire, mais bien une pratique à part entière, qui s'inscrit dans les méthodes « éprouvées » de l'histoire⁹.

Cet accompagnement à la technologie et l'évolution des méthodes en Sciences humaines et sociales, participent en partie, à la condition du succès d'un projet tel que Lectaurep.

8. Rapport, Vie publique/INSEE, 2019, URL : <https://www.vie-publique.fr/en-bref/271657-fracture-numérique-lillectronisme-touche-17-de-la-population>

9. Franziska Heimburger et Émilien Ruiz, « Faire de l'histoire à l'ère numérique : retours d'expériences », *Revue d'histoire moderne contemporaine*, 58-4bis-5 (2011), p. 70-89, URL : <https://www.cairn.info/revue-d-histoire-moderne-et-contemporaine-2011-5-page-70.htm> (visité le 22/09/2020)

Annexes

Les annexes présentent à la fois les livrables réalisés durant le stage et des compléments à ce mémoire. Cette partie contient également les chemins de localisation des fichiers. Ils sont reproduits sur une clé usb et sur un dépôt *Github* accessible à l'adresse suivante : https://github.com/Lucaterre/L-TERRIEL_memoireDeStage_M2TNAH_ENC.

Annexe A

Sources et Ecosystème Lectaurep

localisation : /A-Sources_et_Ecosystème_Lectaurep/ contenant :

A.1 histoire du projet lectaurep

localisation : /A1-histoire_projet_lectaurep/ contenant :

- demande_ocr_Ollion.pdf (pdf reproduit ci-dessous)

A.2 Extraits du corpus des répertoires de notaires

localisation : A2-Extraits_du_corpus_des_répertoires_de_notaires/ contenant :

- exemple_minute.jpg ou Figure A.1
- repertoire_structure_tableaux.png ou Figure A.2

A.3 Outils généraux utilisés dans Lectaurep

localisation : A3-Outils_de_Lectaurep contenant :

- Interface_eScriptorium.png ou Figure A.3
- sharedocs.png ou Figure A.4
- blog_lectaurep.png ou Figure A.5

Paris, le 7 novembre 2005

Note à l'attention de Monsieur François Merlin

Chef du SNTIC

S/c de Monsieur Gérard Ermisse

Directeur du Centre historique des Archives nationales

Michel Ollion

MC

OBJET : Expérimentation d'un accès automatique au contenu d'un répertoire de notaire numérisé

La presse professionnelle (*Lettre des archivistes*, n° 77, septembre-octobre 2005) a fait récemment état d'un nouveau système d'accès automatique au contenu de manuscrits numérisés, installé dans la salle de lecture des Archives départementales des Yvelines. Ce service a fait numériser les registres matricules militaires du XIX^e siècle, ce qui représente 450 000 pages ou images. Un logiciel, mis au point par l'équipe IMADOC de l'IRISA de Rennes (Institut de recherche en informatique et systèmes aléatoires), permet de reconnaître automatiquement les patronymes manuscrits et d'accéder directement aux pages qui intéressent le lecteur, sans qu'il ait été nécessaire d'indexer préalablement les documents. De plus, une plate-forme d'annotation collective offre la possibilité aux lecteurs d'ajouter à un document donné des renseignements supplémentaires utilisables par d'autres chercheurs.

Le Minutier central des notaires de Paris mène actuellement une importante opération d'indexation des images des répertoires de notaires (environ 1 million d'images). Cette indexation permettra de visualiser automatiquement sur écran les pages du répertoire d'un notaire à une date donnée, ce qui représentera un progrès considérable par rapport au système actuel de consultation des microfilms. En revanche, elle ne permettra pas d'accéder directement aux noms des clients. Or, il nous semble utile de faire bénéficier ces documents, très riches en informations sur les personnes et les biens, des progrès actuels des techniques en matière de reconnaissance des écritures manuscrites.

Les pages de répertoires de notaires du XIX^e siècle sont dotées de cases préimprimées qui structurent les informations manuscrites, à l'instar des pages des registres matricules du XIX^e siècle. Compte tenu de cet atout et du fait que la numérisation de ces documents est en voie d'achèvement, nous souhaiterions qu'une expérimentation soit réalisée sur quelques répertoires de notaire du XIX^e siècle. Cette étude préalable devrait servir de test pour envisager la réalisation d'un système de reconnaissance des informations manuscrites consignées sur ces documents.

Une telle opération présenterait l'intérêt d'enrichir et de faire progresser le programme NOEMI qui concerne l'informatisation des instruments de recherche du Minutier central.

Je vous remercie de la suite que vous voudrez bien donner à cette demande.

Françoise MOSSER
Conservateur général
chargé du Minutier central des notaires de Paris

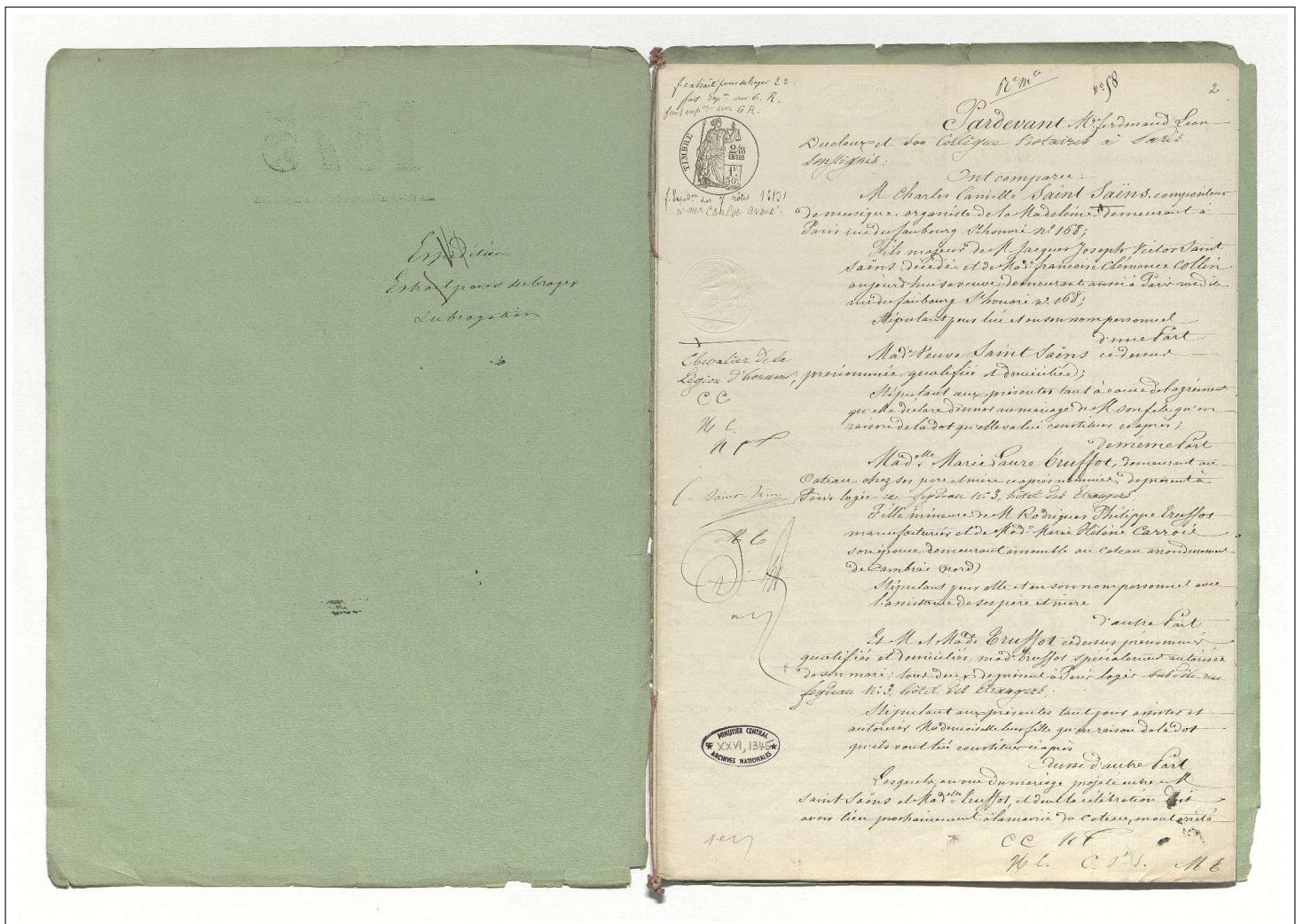


FIGURE A.1 – Exemple de minute notariale ©Archives nationales/DMC, Minute « Contrat de mariage entre Charles Camille Saint-Saëns, compositeur de musique, organiste de la Madeleine demeurant au 168 rue du Faubourg Saint-Honoré, et Marie-Laure Truffot, fille de Rodrigue Truffot, manufacturier au Cateau-Cambrésis », 18 janvier 1875, MC/ET/XXVI/1345 (cote originale), MC/RS//872, lien vers la SIV : https://www.siv.archives-nationales.culture.gouv.fr/siv/UD/FRAN_IR_041418/c1p6uqwjl1o3-xlv0cdl5wlvgv(consulté le 14/09/2020).

*en-tête imprimée
(ignorée)*

Longueville Guillet

N° de répertoire	DATES	NATURE ET ESPÈCE DES ACTES :	NOMS, PRÉNOMS ET DOMICILES DES PARTIES		RELATION de l'Enregistrement.
			EN ACTES	EN PATES	
413	19	Concession			date (année, mois) An 1920 , mois de Février
414	date 15 x- 1919	Résiliation			chiffres / nombres
415	date 1-31	Signature			montant des droits
416	20	Collation			date d'enregistrement (jour) 1-31
417	20	Collevage			3.75
418	20	Déposition			
419	20	Demande			
420	date 30 sept 1919	Discharge			
421	20	Dépôt			
422	20	Demande			
423	20	Organization			
424	20	Retour			
425	20	Prélevage			
426	20	Salgarde			
427	20	Signature			
428	20	Concession d'aut.			
429	20	Contrat			

FIGURE A.2 – Structuration en tableaux des répertoires ©BONHOMME (Marie-Laurence), *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris*, Mémoire de recherche, École nationale des chartes, 2018, pp. 27.

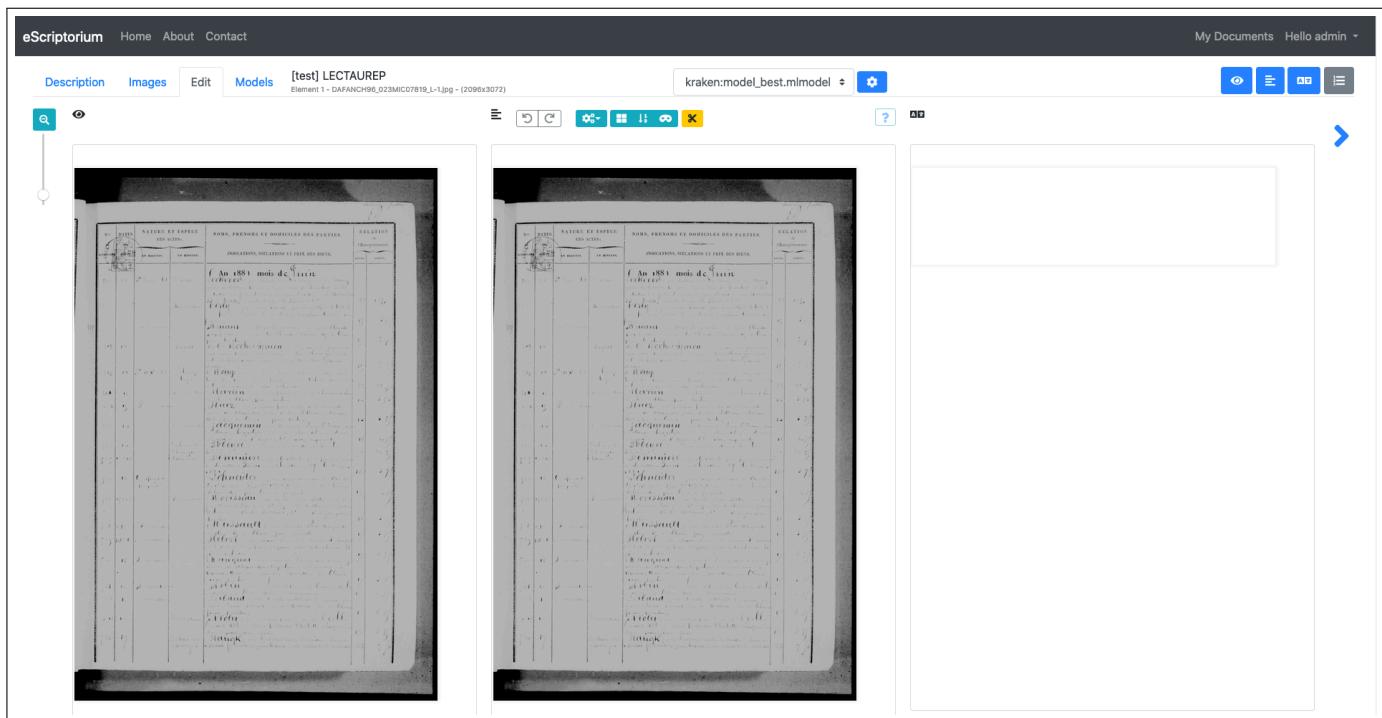


FIGURE A.3 – Application web eScriptorium ©L. Terriel, 2020, eScriptorium

FIGURE A.4 – Exemple du *Golden Set* et du *Random Set* stockés sur l'espace *Sharedocs* (Huma-num) ©L. Terriel, 2020, *Sharedocs* (Huma-num)

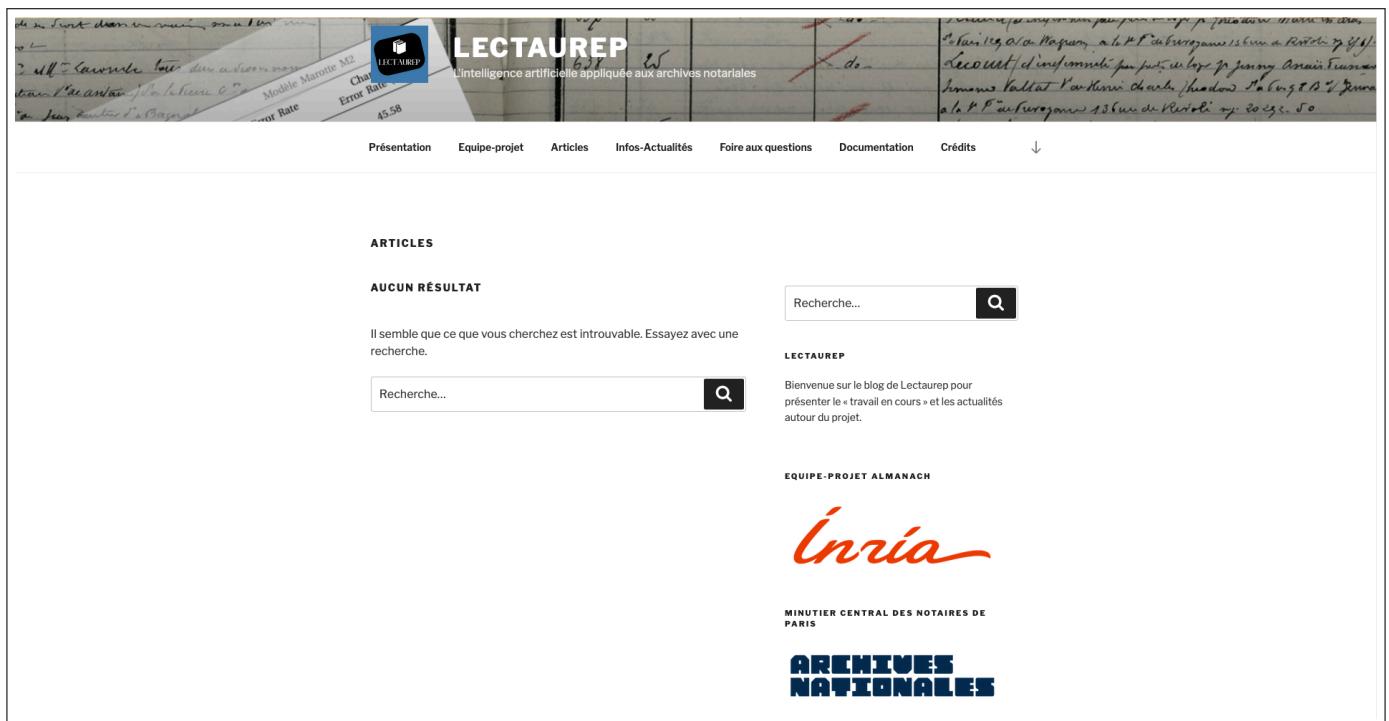


FIGURE A.5 – Blog *hypotheses.org* Lectaurep ©L. Terriel, blog *hypotheses.org* Lectaurep

Annexe B

Format pivot XML-TEI Lectaurep

localisation : /B-Format_pivot_XML_TEI_Lectaurep contenant :

/B-Format_pivot_XML_TEI_Lectaurep/
|__ Doc/.... Regroupe la documentation sur le projet du fichier pivot XML-TEI pour
|__ Lectaurep
| |__ Modélisation_et_documentation_format_pivot/
| | |__ template_pivot_TEI_lectaurep.xml..... Canevas (*template*)
| | | pour formaliser les attentes et les réflexions des acteurs pour l'inclusion des
| | | données et des métadonnées Lectaurep dans le fichier pivot XML-TEI.
| | |__ Les_ODD_aux_formats : XML, PDF et HTML.. La documentation standard
| | | TEI pour le fichier pivot XML-TEI Lectaurep.
| | |__ oddbyexample.xsl.... Une feuille de transformation XSL pour générer une
| | | nouvelle ODD à partir de la *template* XML-TEI pivot.
| |__ Crosswalks_vers_TEI.... Un ensemble de transformations XSL utiles vers les
| | | spécifications TEI et ALTO, issues de différents projets.
| | | |__ ALTO -> TEI
| | | |__ EAD -> TEI (2 versions)
| | | |__ PAGE -> ALTO
| | | |__ PAGE -> TEI
|__ Generator_Lectaurep2TEI/..... CLI Python permettant de simuler
| | la conversion des données issues de Lectaurep (EAD-EAC, ALTO, EXIF) vers un
| | fichier XML-TEI Pivot suivant les recommandations de l'ODD (voir le readme.md
| | pour plus de détails).
| |__ Output/ Dossier dans lequel est générée la sortie du *script*.
| | |__ test_legay_teis.xml Exemple de fichier pivot TEI Lectaurep de test généré
| | en sortie du *script*.

Sets_test_Legay/ Contient un ensemble de fichiers correspondant aux données fournies par Lectaurep pour tester le programme *generator Lectaurep2TEI* et générer une première version du format pivot.

- Data_xml_alto/ Fichiers XML ALTO correspondants aux transcriptions vérité terrain récupérées sur l'espace *ShareDocs* relatives aux images numérisées du répertoire de notaire d'Ernest Legay (étude XXIII).
- Data_xml_ead_eac/. Fichiers XML EAD et XML EAC-CPF correpondants aux instruments de recherche des répertoires du notaire Ernest Legay (étude XXIII) et notices producteurs récupérées sur la Salle des inventaires virtuels des Archives nationales.
 - FRAN_IR_041698.xml ... IR « Minutes et répertoires du notaire Ernest LEGAY, 25 février 1875 - 14 mai 1902 (étude XXIII) ».
 - FRAN_IR_051379.xml ... IR « Images des répertoires du notaire Ernest Legay pour l'étude XXIII ».
 - FRAN_NP_010150.xml Notice producteur de l'étude XXIII.
 - FRAN_NP_011490.xml Notice producteur de Legay, Ernest.
 - Schémas XML et DTD EAD et EAC-CPF
- images/. Images numérisées du répertoire de notaire d'Ernest Legay (étude XXIII) issues du *Golden Set* de l'espace *ShareDocs*.

generator_utils/ Ensemble des modules Python utiles au fonctionnement du CLI *generator Lectaurep2TEI*. Pour les détails des fonctions consulter les *docstrings* dans les *scripts*.

- __init__.py
- build_utils.py
- extract_utils.py
- validation_utils.py

pack_schemaRNG/. Doit accueillir à terme les schémas de validation Relax NG du fichier pivot XML-TEI Lectaurep.

- tei_all.rng Schéma Relax NG *TEI ALL*.

Lectaurep_ALTO2TEI.xsl..... Une feuille de transformation XSL ALTO vers TEI nécessaire pour le fonctionnement du *script* principal. Pour plus de détails consulter la *docstring* de la feuille.

Lectaurep_EADEAC2TEI.xsl... Une feuille de transformation XSL EAD/EAC vers TEI nécessaire pour le fonctionnement du *script* principal. Pour les usages consulter la *docstring*.

- catalog_alto.xml Fichier automatiquement créé par le *script* principal, nécessaire pour l'usage de la fonction Xpath 2.0 `collection()` pour la feuille XSL `Lectaurep_ALTO2TEI.xsl`
- catalog_ead_eac.xml.Lectaurep_ALTO2TEI.xsl...Fichier automatiquement créé par le *script* principal, nécessaire pour l'usage de la fonction Xpath 2.0 `collection()` pour la feuille XSL `Lectaurep_EADEAC2TEI.xsl`
- generator_Lectaurep2TEI_logo.png
- inr_logo_grisbleu.png
- main.py *Script* Python principal d'exécution du CLI *generator Lectaurep2TEI*.
- readme.md Documentation pour installer et lancer le programme.
- requirements.txt...Ensemble des *packages* Python nécessaires à l'utilisation du CLI
- snap_generator.png

Annexe C

Application *Kraken Benchmark*

localisation : /C-Application_Kraken_Benchmark contenant :

```
/C-Application_Kraken_Benchmark/
    Documentation-Reasearch/ ..... Contient les versions du
        notebook Jupyter exposant les réflexions sur les métriques et les algorithmes utilisés
        dans l'application Kraken-Benchmark.
        Evaluation de la similarité entre deux séquences dans le contexte de
            la reconnaissance automatique de caractères..... Notebook Jupyter en
            versions PDF, HTML et IPYNB (format natif).
        Ensemble d'images rattachées au notebook Jupyter
    KB-app/ ..... Dossier contenant les fichiers pour faire fonctionner l'application
        Kraken-Benchmark.
        STS_Tools/ ..... Le package Python Sequences to Similarity créé pour
            l'application Kraken-Benchmark contient deux modules Python.
            STSig.py ..... Module Python Sequences To
                Signals (en cours de développement) ; module pour créer des visualisations
                et des métriques expérimentales pour comparer deux chaînes de caractères.
            SynSemTS.py ..... Module Python Syntactic
                Semantic To Similarity qui contient la plupart des métriques (syntaxiques
                et sémantiques) et les visualisations associées ; utilisé dans l'application pour
                l'analyse de deux chaînes de caractères correspondant à la vérité terrain et
                la transcription issue du système HTR.
            __init__.py
    kb_report/. Dossier contenant les fichiers pour la gestion de la partie affichage
        dans le navigateur de l'application via le package Flask.
        static/. Contient les images de l'application à afficher dans le navigateur.
        templates/. ..... Contient les pages HTML de l'application.
        __init__.py
```

└── **routing.py** *Script Python qui contient les différentes routes URL de l'application.*

└── **kb_utils/**. Dossier contenant un module Python nécessaire au fonctionnement de l'application.

└── **__init__.py**

└── **kb_utils.py** *Module Python contenant des fonctions utiles au fonctionnement de l'application.*

└── **environment.yml**. Fichier pour la création d'un environnement virtuel *Conda* contenant les *packages* Python nécessaires.

└── **kraken_benchmark.py** *Script principal pour l'exécution de l'application Kraken-Benchmark.*

└── **sets_test/**.. Jeux de fichiers pour effectuer des tests dans *Kraken-Benchmark* et *scripts* Python complémentaires.

└── **jules_verne_set_test/**.. Jeux de données utilisés pour les tests fonctionnels de l'application.

└── **images/**.. Contient des numérisations (formats *.jpeg*) de l'ouvrage *Voyage au centre de la terre* récupérés sur *Gallica*.

└── **dataset_GT/**.... Contient les vérités terrains en format texte brut utilisées pour comparer les résultats issus de l'HTR de *Kraken-Benchmark*. Édités à partir du CLI Kraken.

└── **model/**.. Contient le modèle (format *.mlmodel*) entraîné sur le CLI Kraken pour réaliser l'OCR dans *Kraken-Benchmark* sur le *set* de numérisations de *Voyage au centre de la terre*.

└── **sets_tests_lectaurep/**..... Jeux de données utilisés pour tester les modèles de transcription issus du CLI Kraken sur des images de répertoires de notaires sélectionnées pour leurs particularismes (pour plus de détails sur les sets d'images et les modèles HTR utilisés voir le fichier *CR_tests_lectaurep_KB.md*) pour évaluer la qualité des transcriptions. pour plus de précisions, Cf. section 7

└── **different_control_set/**. Contient les transcriptions vérité terrains (GT et les images).

└── **homogeneous_control_set/**. Contient les transcriptions vérité terrains (GT et les images).

└── **set_material_defects/**.. Contient les transcriptions vérité terrains (GT et les images).

└── **set_writing_defects/**... Contient les transcriptions vérité terrains (GT et les images).

screenshots/... Contient des captures de la fonctionnalité *versus text* (comparaison de la vérité terrain et de la transcription HTR dans *Kraken-Benchmark*).

models/..... Contient les modèles HTR, entraînés avec le CLI *Kraken*, et utilisés sur les différents jeux de données.

CR_tests_lectaurep_KB.md..... Compte-rendu présentant le déroulement des tests, la description des sets de tests, des modèles et des résultats des expériences.

details_data_average_tests_model_test_lectaurep_bin_accuracy_6064.mlmodel
- Feuille 1.pdf Fichier PDF présentant la moyenne des résultats des tests avec le modèle 6064, utilisé pour le graphique radar.

details_data_average_tests_model_test_lectaurep_bin_accuracy_8164.mlmodel
- Feuille 1.pdf Fichier PDF présentant la moyenne des résultats des tests avec le modèle 8164, utilisé pour le graphique radar.

radar_test_II_model_0-8164.png Graphique du model 8164.

radar_test_I_model_0-6064.png Graphique du model 6064.

alto2text.py Script Python adapté, provenant de la *Staatsbibliothek Berlin* permettant la conversion de certaines transcriptions vérités terrains en XML ALTO vers des fichiers texte brut.

radar_graph.py Script Python adapté permettant la génération d'un graphique radar, utile pour le rapport concernant les tests spécifiques à Lectaurep durant le stage.

README.md..... Présentation et documentation principale de l'application pour l'installation et l'utilisation de *Kraken-Benchmark*.

ci-test.sh.. Script Bash (Alix Chagué) pour configurer les tests de l'application (avec le logiciel *Pylint*) doit accueillir l'appel des tests unitaires.

.pylintrc..... Fichier pour configurer *Pylint*.

requirements.txt.. Fichier d'installation des *packages*, utilisé exclusivement par le script *ci-test.sh*.

Annexe D

Documents de travail pour le fichier pivot XML-TEI Lectaurep

localisation : /D-Doc_travail_pivot_teilectaurep contenant :

- modèle_metadonnées_vise_V4.png ou Figure D.1
- Generateur-XML-TEI.png ou Figure D.2

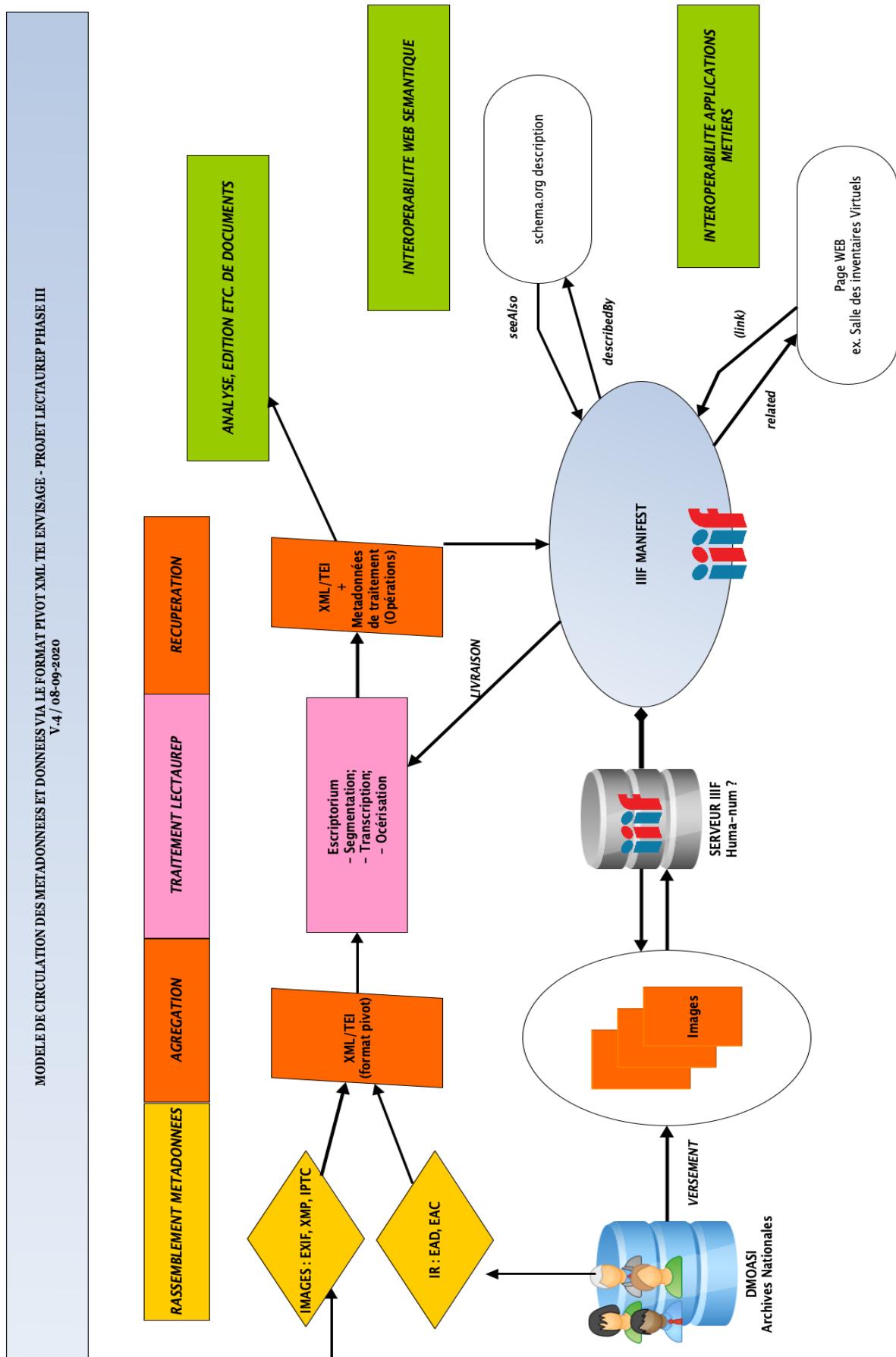


FIGURE D.1 – Schématisation du modèle de circulation des données dans *eScriptorium* souhaité par Lectarep à terme ©L. Terriel, 2020, yEd

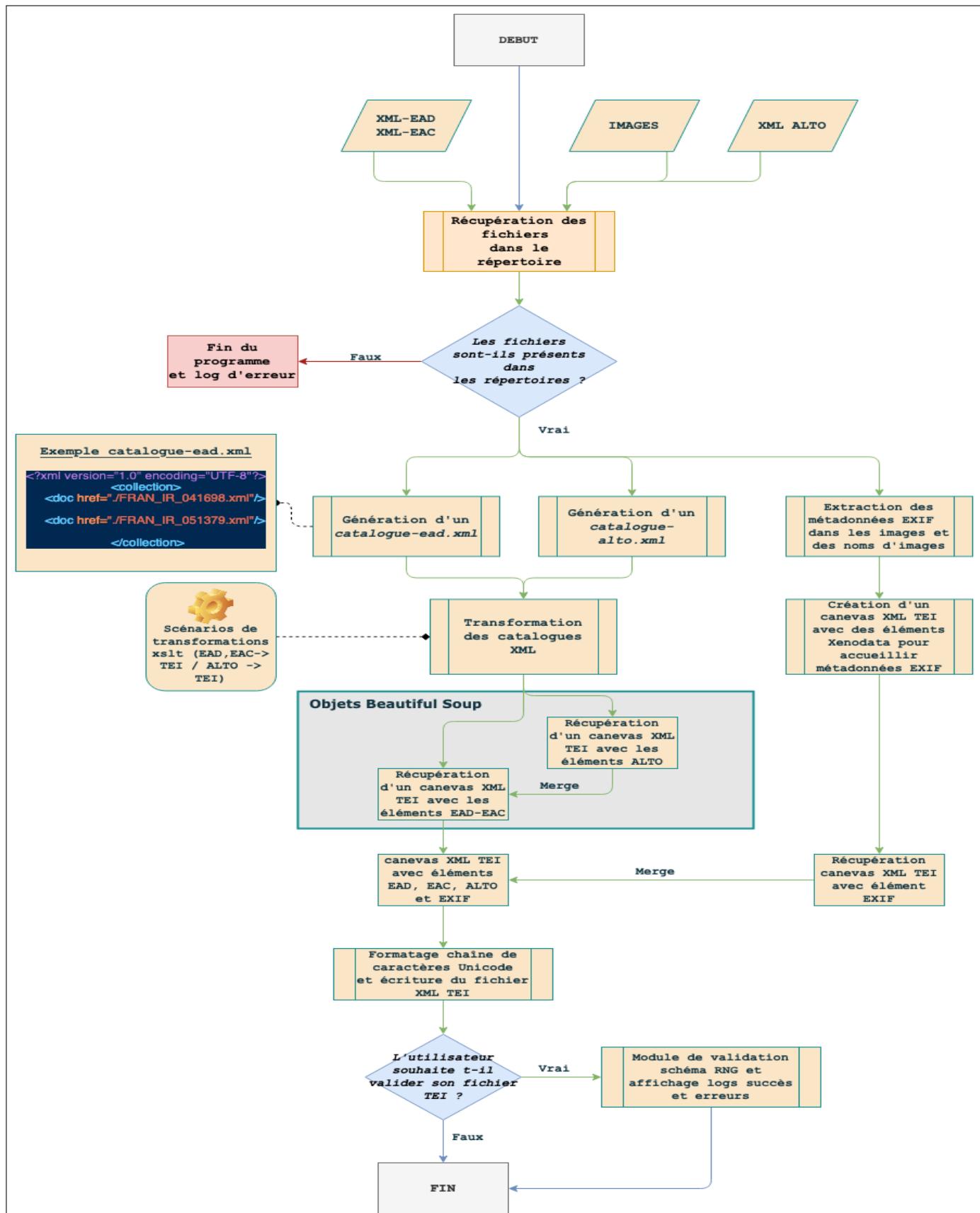


FIGURE D.2 – Algorigramme du programme *Generator Lectaurep-TEI* pour simuler la visualisation d'un fichier XML-TEI pivot comprenant des données provenant de fichiers XML ALTO, EAD et EAC-CPF et des métadonnées EXIF provenant d'images. ©L. Terriel, 2020, Diagrams.net

Annexe E

Documents de travail pour *Kraken-Benchmark*

localisation : /E-Doc_travail_KB contenant :

- detail-model-transkribus.png ou Figure E.1
- compare-texts-transkribus.png ou Figure E.2
- Kraken-Benchmark_modelisation.png ou Figure E.3

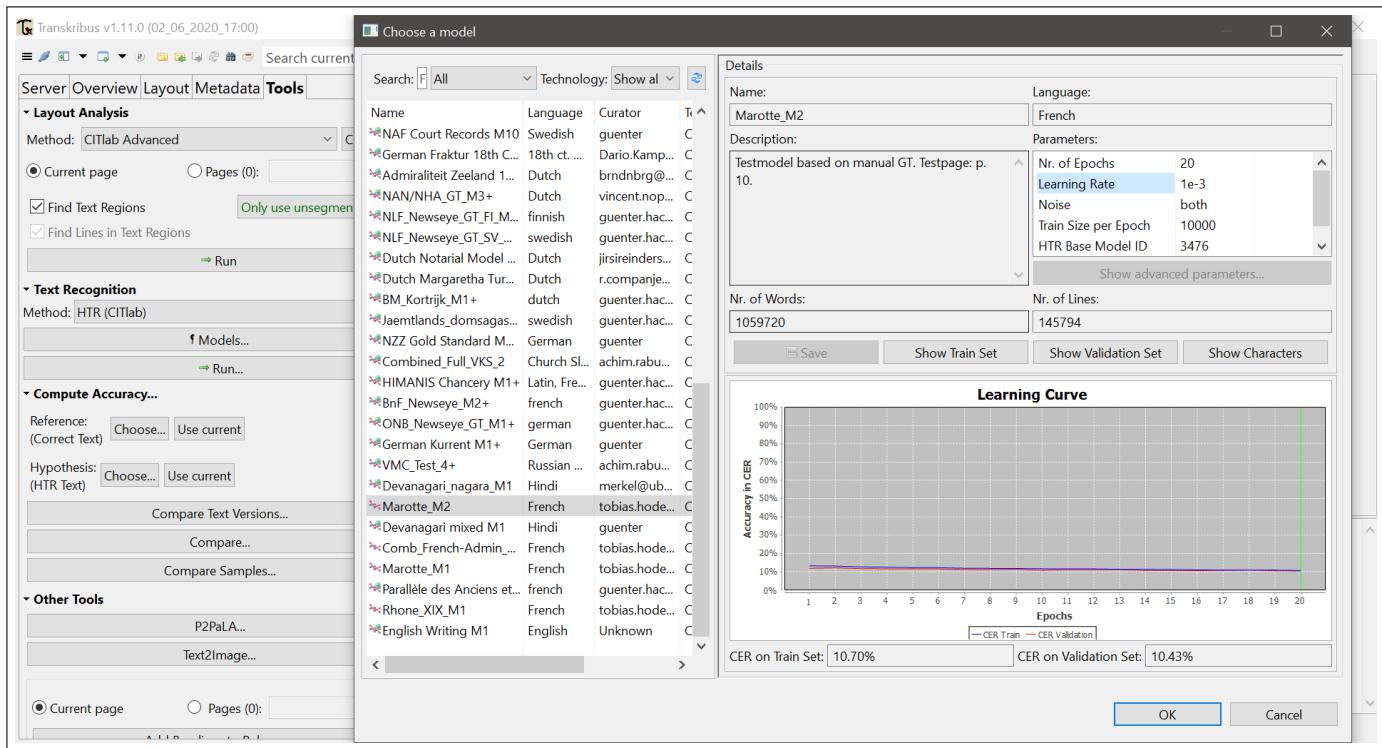


FIGURE E.1 – Fenêtre pour visualiser des détails concernant le modèle envoyé dans l’interface *Transkribus* ©A. Chague, 2020, *Transkribus*

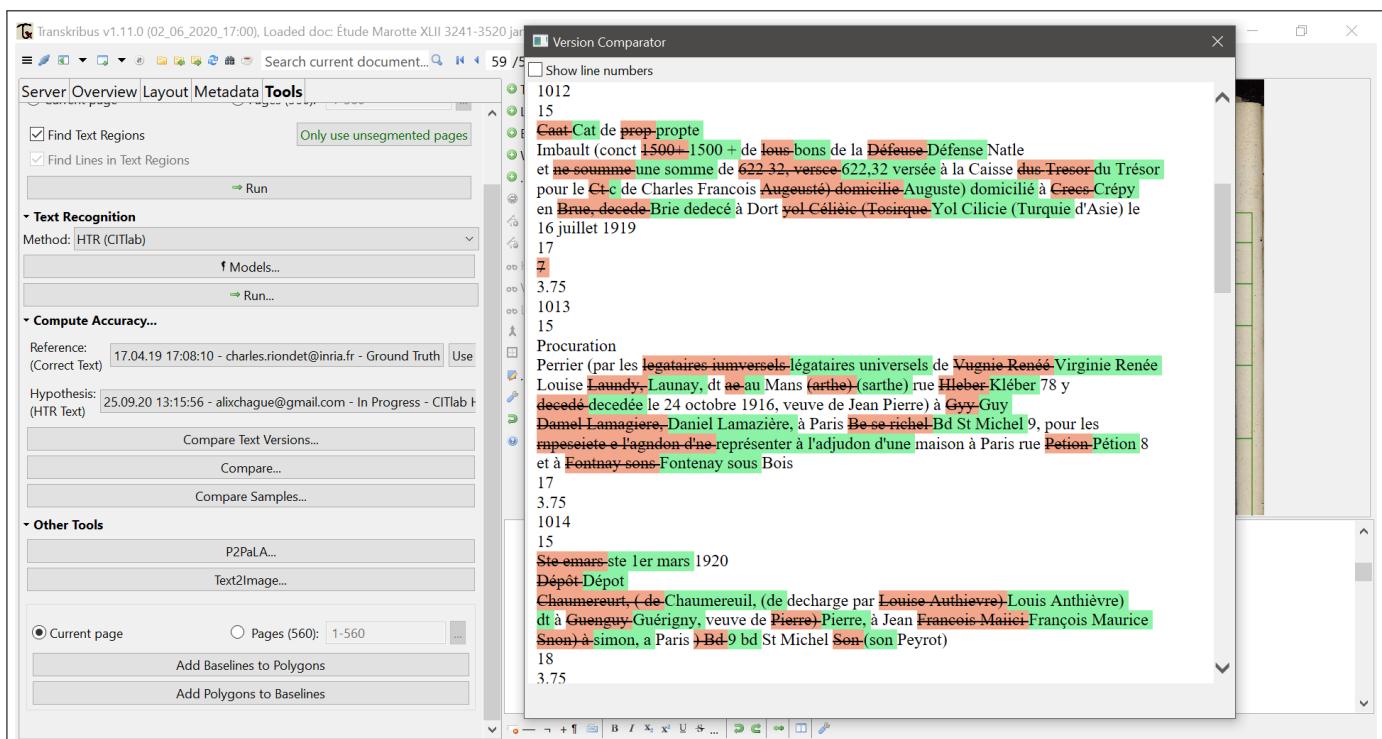


FIGURE E.2 – Fenêtre pour comparer la référence et la prédiction dans l’interface *Transkribus* ©A. Chague, 2020, *Transkribus*

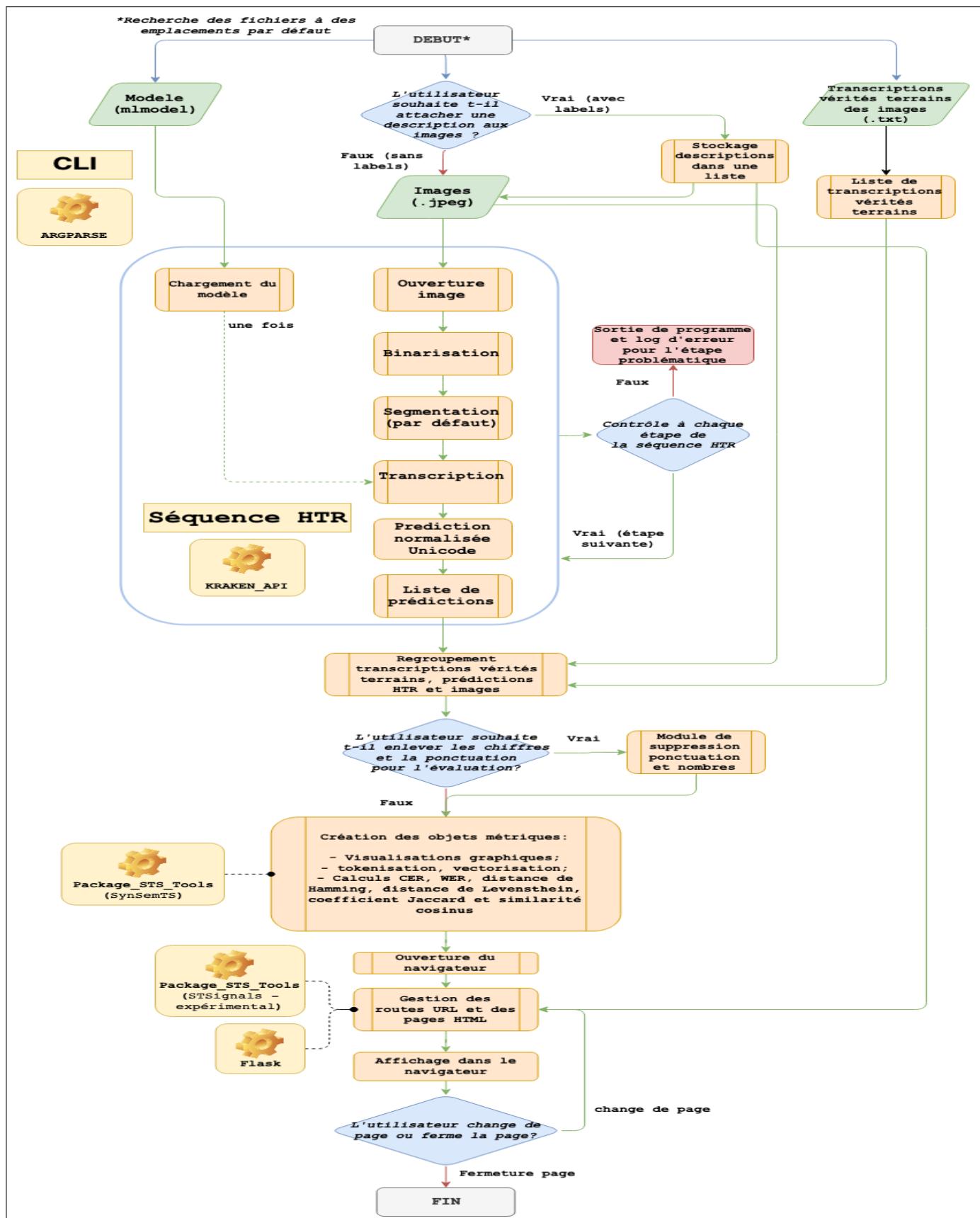


FIGURE E.3 – Algorigramme de Kraken-Benchmark. ©L. Terriel, 2020, Diagrams.net

Annexe F

Scripts Python complémentaires

localisation : /F-scripts_python_comp contenant :

- script_python_POS.py ou Figure F.1
- script_python_exif.py ou Figure F.2
- script_calc.py ou Figure F.3

```

1 """
2 Exemple de script pour l'étiquetage morpho-syntaxique (part-of-speech)
3
4 Auteur : Lucas Terriel
5 Date : 14/09/2020
6 """
7
8 # On importe le package spacy (tâches NLP)
9 import spacy
10 from spacy import displacy
11
12 # On charge un modèle français
13 # ! préalable télécharger le modèle (réseau convolutionnel entraîné sur deux
14 #     corpus, WikiNER et Sequoia) !
15 # python -m spacy download fr_core_news_sm
16 model_fr = spacy.load("fr_core_news_sm")
17
18 # On défini une phrase de test
19 test = "Paul Charles Claude demeurant à Paris avec sa femme"
20
21 def return_POS(sentence):
22     """
23         fonction pour tokeniser la phrase et retourner les étiquettes grammaticale
24         de chaque token.
25
26         :param sentence: phrase
27         :type sentence: str
28         :return: token et étiquettes POS
29         :type return : list
30     """
31
32     # Découpage de la phrase en mots (tokens)
33     document = model_fr(sentence)
34     # Retourne dans un dictionnaire les tokens (X) -clés- et leurs étiquettes
35     # (X.pos_) -valeurs- à partir d'une liste en compréhension
36     return [{(X, X.pos_)} for X in document]
37
38 # affichage du dictionnaire
39 print(return_POS(test))
40
41 # Pour la visualisation dans le navigateur du POS on défini des paramètres de
42 # style
43 options = {"color": "red", "font": "Source Sans Pro"}
44
45 # Visualisation POS dans le navigateur
46 doc = model_fr(test)
47 displacy.serve(doc, style="dep", options=options)

```

FIGURE F.1 – *Script* Python pour effectuer de l'étiquetage morpho-syntaxique (POS) ©L. Terriel, 2020

```
1 """
2 Exemple de script pour afficher les métadonnées EXIF d'une image
3
4 Auteur : Lucas Terriel
5 Date : 14/09/2020
6 """
7
8 # import du module pyExifTool pour l'extraction de métadonnées Exif
9 import exiftool
10
11 # Création de l'objet ExifTool, on localise l'image et récupération des
12 # métadonnées dans un dictionnaire (possibilité de traiter en lot - batch)
13 with exiftool.ExifTool() as et:
14     metadata = et.get_metadata('..../static/Voyage_au_centre_de_la_
15     [...]Verne_Jules_btv1b8600259v_16.jpeg')
16
17 # Affichage des métadonnées à partir du dictionnaire
18 for key, value in metadata.items():
19     print(f'{key} => {value}')
```

FIGURE F.2 – *Script* Python pour afficher les métadonnées Exif d'une image ©L. Terriel, 2020

FIGURE F.3 – *Script Python pour illustrer les différents scores de similarité syntaxique entre deux chaînes de caractères et leurs implémentations* ©L. Terriel, 2020

```

1 """
2 Script pour illustrer les différents scores de similarité syntaxique
3 entre deux chaînes de caractères et leurs implémentations en Python :
4
5 * Algorithme Ratcliff/Obershelp
6 * Distance de Levenshtein
7 * Word Error Rate (Taux d'erreur de mots)
8 * Character Error Rate (Taux d'erreur de mots)
9 * Word Accuracy (Taux de reconnaissance de mots)
10
11 ! Voir les résultats à la fin du script !
12
13 Auteur : Lucas Terriel
14 Date : 14/09/2020
15 """
16
17 # On importe le module NLTK pour découper les phrases en mots et en caractères
18 from nltk import regexp_tokenize
19
20 # On importe le module built-in implémentant l'algorithme Ratcliff/Obershelp
21 import difflib
22
23 # On importe la fonction _fast_levenshtein() de l'API Kraken (module
24     lib.dataset)
24 from kraken.lib.dataset import _fast_levenshtein
25
26 # On définit une phrase de référence (Ground Truth) et une prédiction
27     (transcription HTR) de test
28
28 reference = "En l'an 1920 par la procuration"
29
30 prediction = "En l'an 1920 par le procureur"
31
32 # ! Pré-traitements textuels de la phrase de référence (pour le WER et le CER)
33     !
34
34 def tokenize_words_char(sequence:str) -> list:
35     """
36         * découpage en mots (regex : \W)
37         * découpage en lettres (regex : ')
38
39         sequence (str) : séquence à tokeniser
40         return (list) : tokens mots ou caractères
41     """
42
42 tok_mots = regexp_tokenize(sequence, pattern='\W', gaps=True)
43 tok_caracteres = regexp_tokenize(sequence, pattern=' ', gaps=True)
44 return tok_mots, tok_caracteres

```

suite du script

```

1 # Tokenisation de la référence et de la prédiction (au niveau des mots et des
2     caractères)
3 reference_tok_mots, reference_tok_caracteres = tokenize_words_char(reference)
4 prediction_tok_mots, prediction_tok_caracteres =
5     tokenize_words_char(prediction)
6
7 # Exemples de découpage pour la phrase de référence
8
9 print(reference_tok_mots)
10 print(prediction_tok_mots)
11
12 >>> ['En', 'l', 'an', '1920', 'par', 'la', 'procuration']
13
14 print(reference_tok_caracteres)
15 print(prediction_tok_caracteres)
16
17 >>> ['E', 'n', ' ', 'l', "'", 'a', 'n', ' ', '1', '9', '2', '0',
18     ' ', 'p', 'a', 'r', ' ', 'l',
19     'a', ' ', 'p', 'r', 'o', 'c', 'u', 'r', 'a', 't', 'i', 'o', 'n']
20
21 # calcul du score de similarité avec l'algorithme Ratcliff/Obershelp
22 similarite_ro = difflib.SequenceMatcher(None, reference, prediction).ratio()
23
24 # calcul de la distance de Levenshtein
25 lev_distance = _fast_levenshtein(reference, prediction)
26
27 # calcul du WER (Word Error Rate)
28 def WER(distance:int, reference_mots :list) -> int:
29     """
30         Calcul du Word Error Rate
31         WER = Distance(reference,prediction) / nombre_total_mots_référence
32         (On donne le résultat en pourcentage et en décimal pour le Word accuracy)
33
34         distance (int) : distance d'édition
35         reference_mots (list) : phrase découpée en mots
36         returns (int) : taux d'erreur de mots (WER) en décimal et en pourcentage
37         """
38
39         resultat = (distance / len(reference_mots))
40         resultat_pourcent = (distance / len(reference_mots)) * 100
41         return resultat, resultat_pourcent
42
43 # Le WER travaille au niveau des mots
44 resultat_wer_dec, resultat_wer = WER(_fast_levenshtein(reference_tok_mots,
45     prediction_tok_mots), reference_tok_mots)

```

suite et fin du script

```

1 # calcul du CER (Character Error Rate)
2 def CER(distance:int, reference_caracteres :list) -> int:
3     """
4         Calcul du Word Error Rate
5         CER = Distance(reference,prediction) / nombre_total_caracteres_référence
6         (On donne le résultat en pourcentage)
7
8         distance (int) : distance d'édition
9         reference_mots (list) : phrase découpée en caractères
10        return (int) : taux d'erreur de caractères en pourcentage (CER)
11    """
12
13    resultat_pourcent = (distance / len(reference_caracteres)) * 100
14    return resultat_pourcent
15
16 # Le CER travail au niveau des caractères
17 resultat_cer = CER(lev_distance, reference_tok_caracteres)
18
19 # calcul du Word Accuracy (Taux de reconnaissance des mots)
20 def W_acc(WER:int) -> int:
21     """
22         Calcul du Word Accuracy (Taux de reconnaissance des mots)
23         W_Acc = (1 - WER) * 100
24         (On donne le résultat en pourcentage)
25
26         WER (int) : résultat du WER en décimal
27         return (int) : résultat du Word accuracy
28     """
29
30     resultat_pourcent = (1-WER) * 100
31     return resultat_pourcent
32
33
34
35 # On affiche les différents scores :
36 print(f'Le score de similarité (Ratcliff/Obershelp) est de : {similarite_ro}')
37 print(f'La distance de Levenshtein est de : {lev_distance}')
38 print(f'Le CER est de : {resultat_cer} %')
39 print(f'Le WER est de : {resultat_wer} %')
40 print(f'Le Word accuracy est de : {resultat_wacc} %')
41
42 >>> Le score de similarité (Ratcliff/Obershelp) est de : 0.8333333333333334
43 >>> La distance de Levenshtein est de : 6
44 >>> Le CER est de : 19.35483870967742 %
45 >>> Le WER est de : 28.57142857142857 %
46 >>> Le Word accuracy est de : 71.42857142857143 %

```

Bibliographie

Histoire quantitative et notariale

- CLOULAS (Ivan), « L'inventaire automatisé des actes notariés : principes d'analyse et résultats d'expérimentation aux Archives Nationales de Paris », *Publications de l'École Française de Rome*, 31–1 (1977), p. 127-131, URL : https://www.persee.fr/doc/efr_0000-0000_1977_act_31_1_2242 (visité le 21/09/2020).
- DAUMARD (Adeline) et FURET (François), « Méthodes de l'Histoire sociale : les Archives notariales et la Mécanographie », *Annales*, 14–4 (1959), p. 676-693, URL : https://www.persee.fr/doc/ahess_0395-2649_1959_num_14_4_2865 (visité le 21/09/2020).
- GIRARD (Alain), « Aries Philippe L'homme devant la mort », *Population*, 33–2 (1978), p. 471-472, URL : https://www.persee.fr/doc/pop_0032-4663_1978_num_33_2_16750 (visité le 21/09/2020).
- LE GENDRE (Romain), *Confiance, épargne et notaires. Le marché du crédit à Saint-Maixent et dans sa région au XVIe siècle*, Mémoires et documents de l'École des chartes, 2020.
- LEGAY (Marie-Laure), FÉLIX (Joël) et WHITE (Eugene), « Retour sur les origines financières de la Révolution française », *Annales historiques de la Révolution française*, 2009–356 (2009), p. 183-201, URL : <http://journals.openedition.org/ahrf/10637> (visité le 21/09/2020).
- LIMON-BONNET (Marie-Françoise) et ÉTIENNE (Geneviève), *Les archives notariales : manuel pratique et juridique*, la Documentation française, Paris, 2013.
- LIMON-BONNET (Marie-Françoise), BÉCHU (Claire), LEFÈVRE (Christian) et MAGNIEN (Agnès), *122 minutes d'histoire. Actes des notaires de Paris XVIe-XXe siècle*, Somogy éditions d'art, 2012.
- NEAL (Larry), « Priceless Markets : The Political Economy of Credit in Paris, 1660-1870. By Philip T. Hoffman, Gilles Postel-Vinay, and Jean-Laurent Rosenthal. Chicago : University of Chicago Press, 2000. Pp. xi, 350. » *The Journal of Economic History*, 62–1 (2002), p. 229-231, URL : <https://www.cambridge.org/core/journals/journal-of-economic-history/article/priceless-markets-the-political-economy-of-credit-in-paris-16601870-by-philip-t-hoffman-gilles-postelvinay-and-jeanlaurent-rosenthal-chicago-university-of-chicago-press-2000-pp-xi-350/9AB5C3BAC86592A70A0435F16CB9C4DF> (visité le 21/09/2020).

POISSON (Jean-Paul), « Histoire des populations et actes notariés », *Annales de Démo-graphie Historique*, 1974–1 (1974), p. 51-57, URL : https://www.persee.fr/doc/adh_0066-2062_1974_num_1974_1_1229 (visité le 21/09/2020).

PONCET (Olivier), « Jean-Yves Sarazin. Bibliographie de l'histoire du notariat français, 1200-1815. Préface de Robert Descimon. Paris : Lettrage Distribution, 2004. In-8, 650 pages », *Bibliothèque de l'École des chartes*, 163–2 (2005), p. 559-560, URL : https://www.persee.fr/doc/bec_0373-6237_2005_num_163_2_463767_t1_0559_0000_3 (visité le 21/09/2020).

SARAZIN (Jean-Yves), « L'historien et le notaire : acquis et perspectives de l'étude des actes privés de la France moderne », *Bibliothèque de l'École des chartes*, 160–1 (2002), p. 229-270, URL : https://www.persee.fr/doc/bec_0373-6237_2002_num_160_1_451095 (visité le 21/09/2020).

Ressources sur les écritures XIX^e siècle

ANDRÉ (Anatole), *Le Livre d'écriture, recueil de modèles d'écriture, avec conseils aux élèves, pouvant servir dans toutes les écoles, et particulièrement dans celles où l'on fait usage du cahier unique (individuel)*, 1898, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k14212562> (visité le 11/09/2020).

BERLINER (Arnold), *Cours complet de tous les genres d'écritures usités en France, dédié à ses élèves*, 1862, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k164353h> (visité le 11/09/2020).

CARSTAIRS (Joseph), *Manuel de calligraphie. Méthode complète de Carstairs dite américaine, ou L'art d'écrire en peu de leçons par des moyens prompts et faciles ...* 1829, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k35396g> (visité le 11/09/2020).

FRÉMONT (E.-L.), *Cahiers manuscrits, recueil de toutes sortes d'écritures lithographiées, pour exercer à la lecture des écritures difficiles*, 1837, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k1162328s> (visité le 11/09/2020).

MAGNÉE (François), *Le parfait calligraphe, ou méthode pour apprendre soi-même à écrire en peu de leçons*, 1828.

MOLLIARD et HINARD, *Méthode pratique et simultanée de lecture, d'écriture et d'orthographe*, 1861, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k6152346g> (visité le 11/09/2020).

PAPI (Marc-André), *Traité de calligraphie théorique et pratique, comprenant : l'expédiée française, nouveau genre d'écriture très-lisible et très-rapide, la ronde, la coulée-bâtarde, la gothique et l'anglaise*, 1865, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k1182750j> (visité le 11/09/2020).

WERDET (Jean-Baptiste), *Innovation : leçons d'écriture simplifiée, par Werdet père...* 1841, URL : <https://gallica.bnf.fr/ark:/12148/bpt6k130683n> (visité le 11/09/2020).

Archivistique

ANGJELI (Anila), CLAVAUD (Florence) et ROUSSEL (Stéphanie), « Représenter en RDF, interconnecter et visualiser en graphe des jeux de métadonnées archivistiques de provenances multiples : un projet de prototype », *Gazette des archives*, 245–1 (2017), p. 157-171, URL : https://www.persee.fr/doc/gazar_0016-5522_2017_num_245_1_5523 (visité le 21/09/2020).

CLAVAUD (Florence) et CHARBONNIER (Pauline), *Records in Contexts aux Archives nationales : enjeux et premières réalisations*, hypotheses.org, 2020, URL : https://f.hypotheses.org/wp-content/blogs.dir/2167/files/2020/02/20200128_3_RiCauxAN_EnjeuxPremieresRealisations.pdf.

CLAVAUD (Florence) et HENRY (Cyprien), *Vers un référentiel national des notaires ?*, forum des archivistes, 2016, URL : <https://fdocuments.fr/document/relier-donnees-referentielnotaireschenryfclavaud-final.html> (visité le 21/09/2020).

GUEIT-MONTCHAL (DIR.) (Lydiane), *Abrégé d'archivistique. Principes et pratiques du métier d'archiviste 4e édition revue et augmentée*, AAF, 2020.

Humanités numériques

BEAUGUITTE (Laurent), « L'analyse de réseaux en sciences sociales et en histoire », dans *Le réseau. Usages d'une notion polysémique en sciences humaines et sociales*, Presses Universitaires de Louvain, 2016, p. 9-24, URL : <https://halshs.archives-ouvertes.fr/halshs-01476090> (visité le 14/09/2020).

CANTEAUT (Olivier), GUYOTJEANNIN (Olivier) et PONCET (Olivier), *Actes royaux et princiers à l'ère du numérique (Moyen Âge Temps modernes)*, PUPPA, 2020, URL : <https://acronavarre.hypotheses.org/2810> (visité le 07/09/2020).

DOUEIHI (Milad), *La Grande conversion numérique. suivi de Rêveries d'un promeneur numérique*, Seuil, 2011.

FARAUT (Vivien), *Les outils de représentation graphique de l'espace relationnel face au secret : le cas des conspirateurs du 19 août 1820*, Les Cahiers de Framespa. Nouveaux champs de l'histoire sociale, 2015, URL : <http://journals.openedition.org/framespa/3233> (visité le 15/09/2020).

FOSSIER (Lucie), VAUCHEZ (André) et VIOLANTE (Cinzio), « Informatique et histoire médiévale. Actes du colloque de Rome (20-22 mai 1975) », *Publications de l'École française de Rome*, 31-1 (1977), URL : https://www.persee.fr/issue/efr_0000-0000_1977_act_31_1 (visité le 14/09/2020).

GILLET (Florence), HENGCHEN (Simon), HOOLAND (Seth Van), SINATRA (Michael) et WILDE (Max De), *Introduction aux humanités numériques : méthodes et pratiques*, De Boeck supérieur, 2016.

HEIMBURGER (Franziska) et RUIZ (Émilien), « Faire de l'histoire à l'ère numérique : retours d'expériences », *Revue d'histoire moderne contemporaine*, 58-4bis-5 (2011), p. 70-89, URL : <https://www.cairn.info/revue-d-histoire-moderne-et-contemporaine-2011-5-page-70.htm> (visité le 22/09/2020).

HENGCHEN (Simon), HOOLAND (Seth van), VERBORGH (Ruben) et WILDE (Max de), « L'extraction d'entités nommées : Une opportunité pour le secteur culturel ? », *I2D : information, données & documents*, 52-2 (2015), p. 70-79, URL : <https://www.cairn.info/revue-i2d-information-donnees-et-documents-2015-2-page-70.htm> (visité le 15/09/2020).

JÉGOU (Laurent), *Potentialités de l'analyse-réseau en histoire médiévale*, COL&MON, 2017, URL : <https://colemon.hypotheses.org/102> (visité le 14/09/2020).

MOUNIER (DIR.) (Pierre), *Read/write book 2 : une introduction aux humanités numériques*, OpenEdition Press, 2012, URL : <https://books.openedition.org/oep/226?lang=fr>.

Projets patrimoniaux et numérique

BONHOMME (Marie-Laurence), *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris*, Mémoire de recherche, École nationale des chartes, 2018.

CHAGUÉ (Alix), *LECTAUREP Lecture Automatique de Répertoires*, Atelier Culture, 2019,
URL : <https://webcache.googleusercontent.com/search?q=cache:SS8LEFv8NJIJ:https://www.culture.gouv.fr/Media/Thematiques/Innovation-numerique/Folder/Atelier-Inria-2018/Lectaurep-lecture-automatique-de-repertoires+&cd=1&hl=fr&ct=clnk&gl=fr&client=firefox-b-d>.

HAMEL (Sébastien), MOUFFLET (Jean-François) et STUTZMANN (Dominique), « La recherche en plein texte dans les sources manuscrites médiévales : enjeux et perspectives du projet HIMANIS pour l'édition électronique », *Médiévaux. Langues, Textes, Histoire*, 73–73 (2017), p. 67-96, URL : <http://journals.openedition.org/medievales/8198> (visité le 16/08/2020).

JEANNENEY (Jean-Noël), *Quand Google défie l'Europe*, Fayard/Mille et une nuits, 2010.

LIMON-BONNET (Marie-Françoise), MOUFFLET (Jean-François) et PIRAINO (Gaetano), « L'innovation numérique : un cercle vertueux pour l'archivistique », *La Gazette des archives*, 254–2 (2019), p. 247-281, URL : <https://www.archivistes.org/Les-Archives-nationales-une-refondation-pour-le-XXIe-siecle>.

ROCHEBOUET (Anne), « Introduction : Le texte médiéval à l'épreuve du numérique », *Médiévaux. Langues, Textes, Histoire*, 73–73 (2017), p. 5-12, URL : <http://journals.openedition.org.proxy.chartes.psl.eu/medievales/8163> (visité le 16/08/2020).

Données et analyses

HOOLAND (Seth Van), *L'application de l'intelligence artificielle au traitement de la bureautique et des mails*, programme du colloque, Colloque "Les Archives au défi du numérique" (17 et 18 octobre 2019), 2019, URL : <https://www.diplomatie.gouv.fr/fr/archives-diplomatiques/action-scientifique-et-culturelle/colloques-et-conferences/article/colloque-les-archives-au-defi-du-numerique-17-et-18-octobre-2019>.

POUPEAU (Gautier), *Visite guidée au pays de la donnée - Du modèle conceptuel au modèle physique*, cours, 2019, URL : <https://fr.slideshare.net/lespetitescases/visite-guide-au-pays-de-la-donnee-du-modle-conceptuel-au-modle-physique> (visité le 13/08/2020).

WIKIPÉDIA, *Donnée (informatique)*, URL : [https://fr.wikipedia.org/w/index.php?title=Donn%C3%A9e_\(informatique\)&oldid=174663903](https://fr.wikipedia.org/w/index.php?title=Donn%C3%A9e_(informatique)&oldid=174663903) (visité le 23/09/2020).

IA, TAL et HTR

ARES OLIVEIRA (Sofia) et KAPLAN (Frederic), *Comparing human and machine performances in transcribing 18th century handwritten Venetian script*, DH2018, 2018, URL : <https://dh2018.adho.org/en/comparing-human-and-machine-performances-in-transcribing-18th-century-handwritten-venetian-script/> (visité le 21/09/2020).

BARANES (Marion), « Vers la correction automatique de textes bruités : Architecture générale et détermination de la langue d'un mot inconnu », dans *RECITAL'2012 - Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, 2012, URL : <https://hal.inria.fr/hal-00701400> (visité le 15/09/2020).

BARRERE (Killian), COÜASNON (Bertrand) et LEMAITRE (Aurélie), *Results of a PyTorch implementation of an Handwritten Text Recognition Framework*, rapport, INTUIDOC Research group, 2018, URL : http://perso.eleves.ens-rennes.fr/people/killian.barrere/papers/Results_of_a_PyTorch_implementation_of_an_Handwritten_Text_Recognition_Framework.pdf.

BLUCHE (Théodore), KERMORVANT (Christopher) et STUTZMANN (Dominique), *Automatic Handwritten Character Segmentation for Paleographical Character Shape Analysis*, A2IA / IRHT, International Workshop on Document Analysis Systems, 2016, URL : http://www.tbluche.com/files/das16_slides.pdf.

BONHOMME (Marie-Laurence), *Répertoire des Notaires parisiens Segmentation automatique et reconnaissance d'écriture*, rapport, Inria, 2018, URL : <https://hal.inria.fr/hal-01949198> (visité le 13/09/2020).

BOROS (Emanuela), TOUMI (Alexis), ROUCHET (Erwan), ABADIE (Bastien), STUTZMANN (Dominique) et KERMORVANT (Christopher), *Automatic Page Classification in a Large Collection of Manuscripts Based on the International Image Interoperability Framework*, International Conference on Document Analysis and Recognition, 2019, URL : <https://www.computer.org/csdl/proceedings-article/icdar/2019/301400a756/1h81wiF3AA0> (visité le 14/09/2020).

CHAUMARTIN (François-Régis) et LEMBERGER (Pirmin), *Le traitement automatique des langues. Comprendre les textes grâce à l'intelligence artificielle*, Dunod, 2020.

- EIKVIL (Line), *OCR - Optical Character Recognition*, 1993, URL : <https://www.nr.no/~eikvil/OCR.pdf>.
- GANASCIA (Gabriel), *Le Mythe de la Singularité*, Seuil, 2017.
- Handwritten Text Recognition Workflow*, Transkribus Wiki, URL : https://transkribus.eu/wiki/index.php/Handwritten_Text_Recognition_Workflow (visité le 11/09/2020).
- KERMORVANT (Christopher), *La reconnaissance d'écriture manuscrite*, Data Analytics Post, 2019, URL : <https://dataanalyticspost.com/la-reconnaissance-decriture-manuscrite-de-nouvelles-applications-pour-un-des-plus-vieux-problemes-dia/> (visité le 11/09/2020).
- KIESSLING (Benjamin), *Kraken - an Universal Text Recognizer for the Humanities*, DH2019, 2019, URL : <https://dev.clariah.nl/files/dh2019/boa/0673.html> (visité le 11/09/2020).
- KIESSLING (Benjamin), SAVANT (Sarah Bowen), ROMANOV (Maxim) et THOMAS MILLER (Matthew), « Important New Developments in Arabographic Optical Character Recognition (OCR) », *CoRR*, abs/1703.09550 (2017), URL : https://www.academia.edu/28923960/Important_New_Developments_in_Arabographic_Optical_Character_Recognition_OCR_ (visité le 13/09/2020).
- LAVRENKO (Victor), RATH (Toni M.) et MANMATHA (R.), « Holistic word recognition for handwritten historical documents », dans *First International Workshop on Document Image Analysis for Libraries*, 2004, URL : <http://ciir.cs.umass.edu/pubfiles/mmm-319.pdf>.
- LE PEVEDIC (Solenn) et MAUREL (Denis), « Retour sur les annotations des entités nommées dans les campagnes d'évaluation françaises et comparaison avec la TEI », *Corela. Cognition, représentation, langage*, 14–2 (2016), URL : <http://journals.openedition.org/corela/4644> (visité le 16/09/2020).
- MAGALLON (Thibault), BÉCHET (Frédéric) et FAVRE (Benoit), « Détection d'erreurs dans des transcriptions OCR de documents historiques par réseaux de neurones récurrents multi-niveau », dans *25e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, Rennes, France, 2018, URL : <https://hal.archives-ouvertes.fr/hal-01905258> (visité le 15/09/2020).
- McCULLOCH (Warren S.) et PITTS (Walter), « A logical calculus of the ideas immanent in nervous activity », *The bulletin of mathematical biophysics*, 5–4 (1943), p. 115–133, URL : <https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf> (visité le 11/09/2020).
- MIOULET (Luc), *Reconnaissance de l'écriture manuscrite avec des réseaux récurrents*, thèse, Université de Rouen, 2015, URL : <https://hal.archives-ouvertes.fr/tel-01301728> (visité le 11/09/2020).

- MÜHLBERGER (Günter), KAHLE (Philip) et COLUTTO (Sebastian), « Preprint : Handwritten Text Recognition (HTR) of Historical Documents as a Shared Task for Archivists, Computer Scientists and Humanities Scholars. The Model of a Transcription Recognition Platform (TRP) » (, 2014), URL : https://www.academia.edu/8601748/Preprint_Handwritten_Text_Recognition_HTR_of_Historical_Documents_as_a_Shared_Task_for_Archivists_Computer_Scientists_and_Humanities_Scholars_The_Model_of_a_Transcription_and_Recognition_Platform_TRP_ (visité le 14/09/2020).
- OLVER (Chris), *Machine learning of an 18th century hand : transcribing the essays of George III*, Georgian Papers Programme, 2017, URL : <https://georgianpapers.com/2017/01/20/machine-learning-18th-century-hand-transcribing-essays-george-iii/> (visité le 21/09/2020).
- OUJI (Asma), *Segmentation et classification dans les images de documents numérisés*, thèse, INSA de Lyon, 2012, URL : <https://tel.archives-ouvertes.fr/tel-00749933> (visité le 11/09/2020).
- PRATIKAKIS (Ioannis), ZAGORI (Konstantinos), KADDAS (Panagiotis) et GATOS (Basilis), « ICFHR 2018 Competition on Handwritten Document Image Binarization (H-DIBCO 2018) », dans *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, URL : https://eee.hbut.edu.cn/_local/9/3C/5B/599ED504CDE84710DC7A07A7ABF_A91E00B3_5F797.pdf?e=.pdf.
- RAYAR (Frédéric), JEAN-YVES RAMEL et JIMENES (Rémi), *Exploiting Document Image Analysis in the Humanities*, 2012, URL : <https://halshs.archives-ouvertes.fr/halshs-00805863> (visité le 14/09/2020).
- ROMARY (Laurent), « Natural Language Processing for Historical Texts Michael Piotrowski (Leibniz Institute of European History) Morgan & Claypool (Synthesis Lectures on Human Language Technologies, edited by Graeme Hirst, volume 17), 2012, ix+157 p », *Computational Linguistics - MIT*, 40-1 (2014), p. 231-233, URL : <https://hal.inria.fr/hal-01016318> (visité le 10/09/2020).
- ROSENBLATT (F.), « The perceptron : a probabilistic model for information storage and organization in the brain. » *Psychological review*, 65-6 (1958), p. 386-408, URL : <https://psycnet.apa.org/record/1959-09865-001>.
- RUIZ (Pablo), *Concept-based and relation-based corpus navigation : applications of natural language processing in digital humanities*, thèse, PSL Research University, 2017, URL : <https://tel.archives-ouvertes.fr/tel-01575167> (visité le 15/09/2020).
- RUSSELL (Stuart) et NORVIG (Peter), *Intelligence artificielle : Avec plus de 500 exercices*, Pearson Education France, 2010.
- SWAILEH (Wassim), *Des modèles de langage pour la reconnaissance de l'écriture manuscrite*, thèse, Université de Normandie, 2017, URL : <http://www.theses.fr/2017NORMR024> (visité le 11/09/2020).

- THOMAS (Simon), CHATELAIN (Clément), PAQUET (Thierry) et HEUTTE (Laurent), « Un modèle neuro markovien profond pour l'extraction de séquences dans des documents manuscrits », *Document numerique*, 16–2 (2013), p. 49-68, URL : <https://www.cairn.info/revue-document-numerique-2013-2-page-49.html> (visité le 11/09/2020).
- TURING (Alan M.), « Computing Machinery and Intelligence », *Mind*, LIX–236 (1950), p. 433-460, URL : <https://academic.oup.com/mind/article/LIX/236/433/986238> (visité le 11/09/2020).
- VARELA (Francisco), *Invitation aux sciences cognitives*, Seuil, 1996.
- VILLANI (Cédric), *Rapport de Cédric Villani : donner un sens à l'intelligence artificielle (IA)*, Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation, 2018, URL : <https://www.enseignementsup-recherche.gouv.fr/cid128577/rapport-de-cedric-villani-donner-un-sens-a-l-intelligence-artificielle-ia.html> (visité le 11/09/2020).
- WIKIBOOKS, *Algorithm Implementation/Strings/Levenshtein distance*, 2020, URL : https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance (visité le 17/09/2020).
- WIKIPÉDIA, *Distance de Levenshtein*, URL : https://fr.wikipedia.org/w/index.php?title=Distance_de_Levenshtein&oldid=170770063 (visité le 12/09/2020).
- *Indice et distance de Jaccard*, URL : https://fr.wikipedia.org/w/index.php?title=Indice_et_distance_de_Jaccard&oldid=162581283 (visité le 11/09/2020).
 - *Matthews correlation coefficient*, URL : https://en.wikipedia.org/w/index.php?title=Matthews_correlation_coefficient&oldid=974404329 (visité le 11/09/2020).
 - *Méthode des k plus proches voisins*, URL : https://fr.wikipedia.org/w/index.php?title=M%C3%A9thode_des_k_plus_proches_voisins&oldid=173680633 (visité le 11/09/2020).
 - *Modèle de Markov caché*, URL : https://fr.wikipedia.org/w/index.php?title=Mod%C3%A8le_de_Markov_cach%C3%A9&oldid=163612626 (visité le 11/09/2020).
 - *N-gramme*, URL : <https://fr.wikipedia.org/w/index.php?title=N-gramme&oldid=167585561> (visité le 23/09/2020).
 - *Précision et rappel*, URL : https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel (visité le 11/09/2020).
 - *Reconnaissance de l'écriture manuscrite*, URL : https://fr.wikipedia.org/w/index.php?title=Reconnaissance_de_l%27%C3%A9criture_manuscrite&oldid=171738712 (visité le 11/09/2020).

- *Reconnaissance optique de caractères*, URL : https://fr.wikipedia.org/w/index.php?title=Reconnaissance_optique_de_caract%C3%A8res&oldid=169225409 (visit   le 11/09/2020).
- *Similarit   cosinus*, Page Version ID : 167624629, URL : https://fr.wikipedia.org/w/index.php?title=Similarit%C3%A9_cosinus&oldid=167624629 (visit   le 11/09/2020).

Langage XML et standards

- BELAÏD (Abdel), FALK (Ingrid) et RANGONI (Yves), « Représentation des données en XML pour l'analyse d'images de documents », *Conférence Internationale sur l'Ecrit et le Document* (, 2007), URL : <http://lodel.irevues.inist.fr/cide/index.php?id=147> (visité le 07/09/2020).
- BURNARD (Lou), « What is TEI Conformance, and Why Should You Care? », *Journal of the Text Encoding Initiative*, 2019-2020-12 (2020), URL : <http://journals.openedition.org/jtei/1777> (visité le 16/09/2020).
- BURNARD (Lou) et BURGHART (Marjorie), *Qu'est-ce que la Text Encoding Initiative?*, OpenEdition Press, 2015, URL : <http://books.openedition.org/oep/1237>.
- CAMPS (Jean-Baptiste), *Structuration des données et des documents : balisage XML - Personnaliser la TEI : One Document Does it all*, cours, 2017, URL : https://halshs.archives-ouvertes.fr/cel-01706530/file/06_TEI_ODD_Camps_20170202.pdf.
- CAPELLI (Laurent), FARHI (Laurence) et ROMARY (Laurent), *A TEI conformant pivot format for the HAL back-office*, Text Encoding Initiative Conference and members meeting 2015, 2015, URL : <https://hal.archives-ouvertes.fr/hal-01221774> (visité le 10/09/2020).
- CROZAT (Stéphane), « Standardisation des formats documentaires pour les chaînes éditoriales d'UNIT : un schéma pivot », *TICE* (, 2006), URL : <https://stph.crzt.fr/res/crozat06tice.pdf> (visité le 07/09/2020).
- HOLMES (Martin), *The XPath collection() function : pulling in multiple documents*, 2013, URL : http://web.uvic.ca/~mholmes/dhoxss2013/handouts/collection_function.pdf.
- INIST/CNRS, *ISTEX et Conditor convertis au format TEI*, INIST, 2020, URL : <https://www.inist.fr/realisations/istex-et-conditor-convertis-au-format-tei/> (visité le 10/09/2020).
- MONELLA (Paolo), *Linking Text and image : TEI XML and IIIF*, 2019, URL : <http://www1.unipa.it/paolo.monella/reires2019/> (visité le 10/09/2020).
- STUTZMANN (Dominique), *EAD-TEI et TEI-EAD : quelques réflexions sur la conversion des notices de manuscrits médiévaux d'un format à l'autre*, Écriture médiévale &

numérique, 2019, URL : <https://oriflamms.hypotheses.org/1715> (visité le 10/09/2020).

Logiciels et services

ALDER (Gaudenz) et BENSON (David), *Diagrams.net (ex-Draw.io)*, URL : <https://www.diagrams.net/>.

CLÉO, *fr.hypotheses.org Blogs en sciences humaines et sociales*, URL : <https://fr.hypotheses.org/> (visité le 10/09/2020).

DUTCH LANGUAGE INSTITUTE (INL), *alto2tei*, URL : <https://github.com/INL/OpenConvert/blob/master/resources/xsl/alto2tei.xsl>.

GOUGELET (Pierre-Emmanuel), *XnView*, URL : <https://www.xnview.com/fr/>.

HAMMERSLEY (John) et LEES-MILLER (John), *Overleaf*, URL : <https://www.overleaf.com/project>.

JETBRAINS, *IDE PyCharm*, URL : <https://www.jetbrains.com/pycharm/>.

KAY (Michael), *Saxon XSLT*, version 10.1. URL : <http://saxon.sourceforge.net/>.

LOPEZ (Patrice), *Entity-fishing - Entity Recognition and Disambiguation*, URL : <http://nerd.huma-num.fr/nerd/>.

MATTIS (Peter) et KIMBALL (Spencer), *GIMP*, URL : <gitlab.gnome.org/GNOME/gimp>.

OTTO (Mark) et THORNTON (Jacob), *Bootstrap*, version 4.5, URL : <https://getbootstrap.com/>.

PRESTON-WERNER (Tom), WANSTRATH (Chris), CHACON (Scott) et HYETT (P. J.), *Github*, URL : <https://github.com/>.

SIJBRANDIJ (Sid), *Gitlab*, URL : <https://gitlab.com/explore>.

SKINNER (Jon), *Sublime Text*, URL : <https://www.sublimetext.com/>.

SYNCRO SOFT LTD. ROMANIA, *Oxygen XML Editor*, URL : <https://www.oxygenxml.com/>.

UNIVERSITY OF OXFORD - BODLEIAN LIBRARY, *ead2enrich*, URL : <http://projects.oucs.ox.ac.uk/ENRICH/XSLT/xsl/ead2enrich.xsl>.

Documentation *packages* Python

argparse, URL : <https://docs.python.org/fr/3/howto/argparse.html> (visité le 18/09/2020).

bs4 - BeautifulSoup 4.4.0, URL : <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.

difflib, URL : <https://docs.python.org/3/library/difflib.html> (visité le 18/09/2020).

Flask 1.1.2, URL : <https://flask.palletsprojects.com/en/1.1.x/>.

glob, URL : <https://docs.python.org/fr/3.6/library/glob.html> (visité le 18/09/2020).

Jinja 2.11, URL : <https://jinja.palletsprojects.com/en/2.11.x/> (visité le 18/09/2020).

Kraken API, URL : <http://kraken.re/api.html>.

lxml 4.5, URL : <https://lxml.de/>.

math, URL : <https://docs.python.org/3/library/math.html> (visité le 18/09/2020).

matplotlib 3.1.3, URL : <https://matplotlib.org/contents.html#>.

nltk 3.5, URL : <https://www.nltk.org/>.

numpy 1.19, URL : <https://numpy.org/doc/stable/>.

os, URL : <https://docs.python.org/fr/2.7/library/os.html> (visité le 18/09/2020).

pandas 1.0.3, URL : <https://pandas.pydata.org/docs/>.

pillow 7.2.0, URL : <https://pillow.readthedocs.io/en/stable/>.

prompt-toolkit 3.0.5, URL : <https://python-prompt-toolkit.readthedocs.io/en/master/>.

PyExifTool 0.1, URL : <http://smarnach.github.io/pyexiftool/>.

pyfiglet 0.8.0, URL : <https://github.com/pwaller/pyfiglet>.

PYTHON SOFTWARE FOUNDATION. PYTHON LANGUAGE REFERENCE, *Python Language Reference*, version 3.7, URL : <https://www.python.org/>.

python-Levenshtein 0.12.0, URL : <https://rawgit.com/ztane/python-Levenshtein/master/docs/Levenshtein.html>.

scikit-learn 0.22.1, URL : https://scikit-learn.org/stable/user_guide.html.

seaborn 0.10.1, URL : <https://seaborn.pydata.org/>.

spaCy, URL : <https://spacy.io/usage/spacy-101>.

termcolor 1.1.0, URL : <https://pypi.org/project/termcolor/>.

tqdm 4.46.1, URL : <https://tqdm.github.io/>.

Werkzeug, URL : <https://werkzeug.palletsprojects.com/en/1.0.x/> (visit  le 18/09/2020).

Utilitaires

CAMPS (Jean-Baptiste), *biblatex-enc*, 2017, URL : <https://github.com/%20Jean-Baptiste-Camps/biblatex-enc>.

DATTA (Dilip), *LaTeX in 24 hours : a practical guide for scientific writing*, Springer, 2017.

Stack Overflow - Where Developers Learn, Share, & Build Careers, URL : <https://stackoverflow.com/> (visité le 10/09/2020).

TEI CONSORTIUM, *P5 : Guidelines for Electronic Text Encoding and Interchange*, 2020, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/fr/html/index.html> (visité le 10/09/2020).

TeX - LaTeX Stack Exchange, URL : <https://tex.stackexchange.com/> (visité le 10/09/2020).

WIKIBOOKS, *LaTeX*, 2016, URL : <https://fr.wikibooks.org/wiki/LaTeX> (visité le 10/09/2020).

Table des figures

1.1	Exemple de répertoire de notaire ©Archives nationales/DMC, MC/RE/X-LIII/42, étude XLIII du notaire Louis Marie Joseph Marotte	17
1.2	Illustration des fonctionnalités en cours de développement durant la phase 3 de Lectaurep ©A. Chagué, 2020, Diagrams.net	22
2.1	Les différentes disciplines de l'IA ©L. Terriel, 2020, Diagrams.net	25
2.2	Exemples de types d'écritures XIX^e siècle rencontrées dans les répertoires de notaires , de gauche à droite et de haut en bas : système d'écriture dit « de Magnée » extrait de MAGNÉE (François), <i>Le parfait calligraphe, ou méthode pour apprendre soi-même à écrire en peu de leçons</i> , 1828, exemple de « d » delta et de « Q » majuscule archaïque repris de MOLLIARD et HINARD, <i>Méthode pratique et simultanée de lecture, d'écriture et d'orthographe</i> , 1861, URL : https://gallica.bnf.fr/ark:/12148/bpt6k6152346g (visité le 11/09/2020), exemples de modules de tracés des jambages et des hampes WERDET (Jean-Baptiste), <i>Innovation : leçons d'écriture simplifiée, par Werdet père...</i> 1841, URL : https://gallica.bnf.fr/ark:/12148/bpt6k130683n (visité le 11/09/2020), d'écriture « coulée » Ancien Régime repris de FRÉMONT (E.-L.), <i>Cahiers manuscrits, recueil de toutes sortes d'écritures lithographiées, pour exercer à la lecture des écritures difficiles</i> , 1837, URL : https://gallica.bnf.fr/ark:/12148/bpt6k1162328s (visité le 11/09/2020), exemples de gothique moderne dite « fracture », de cursive anglaise, de ronde minutée, d'écriture latine et de taille de plume extraits de BERLINER (Arnold), <i>Cours complet de tous les genres d'écritures usités en France, dédié à ses élèves</i> , 1862, URL : https://gallica.bnf.fr/ark:/12148/bpt6k164353h (visité le 11/09/2020).	30
2.3	Quelques exemples de transcriptions réalisées par des annotateurs de Lectaurep dans <i>eScriptorium</i> . ©Captures fournies par A. Rostaing (AN/DMC), 2020, <i>eScriptorium</i>	33

2.4	Illustration des époques (<i>stage</i>) d'apprentissage pour la création d'un modèle de transcription avec Kraken ©L. Terriel, 2020, Kraken/cluster de calcul INRIA-RIOC	36
2.5	La machine à lire de Tauschek. Premier système OCR électro-mécanique. ©Patent Fetcher	38
2.6	Illustration simplifiée de la <i>méthode des k plus proches voisins</i> (k-NN) ©L. Terriel, 2020, Diagrams.net	39
2.7	Illustration simplifiée d'un neurone formel ©L. Terriel, 2020, Diagrams.net	43
2.8	Illustration simplifiée d'un réseau de neurones à deux couches de neurones cachées ©L. Terriel, 2020, Diagrams.net	44
2.9	Architecture hybride des différentes couches disposant de RNN (BiLSTM) et de neurones à convolution, utilisée dans <i>Kraken</i> ©KIESSLING (Benjamin), <i>Kraken - an Universal Text Recognizer for the Humanities</i> , DH2019, 2019, URL : https://dev.clariah.nl/files/dh2019/boa/0673.html (visité le 11/09/2020)	45
2.10	Exemple d'une tokenisation en mots réalisée avec un tokenizer développé à partir des expressions régulières (Regex) et le tokenizer du package Python NLTK ©L. TERRIEL, 2020, <i>Pycharm</i>	47
2.11	Extrait d'un répertoire du notaire Marotte passé dans l'outil <i>Entity-fishing</i> pour l'étiquetage des entités nommées et sa réponse en JSON comprenant le taux de confiance (<i>weight</i>) accordé pour chaque candidat du référentiel Wikipédia. ©L. TERRIEL, 2020, <i>Entity-fishing</i>	48
2.12	Exemple de <i>Part-of-Speech Tagging</i> réalisé sur un document de vérité terrain du répertoire de notaire Marotte. Réalisé avec un <i>script</i> Python (Cf. Annexes, Figure F.1) ©L. TERRIEL, 2020, <i>Pycharm</i>	49
2.13	Exemple simplifié de représentation vectorielle des mots (<i>Word embedding</i>). Les mots « contrat » et « acte » ont des vecteurs proches (vecteurs colinéaires) tandis que le mot « Paris » présente un vecteur éloigné par rapport à ses pairs (vecteurs orthogonaux). ©L. Terriel, 2020, Diagrams.net	49
2.14	Un exemple de représentation en réseaux sous la forme d'un graphe entre acteurs du complot (suivant les charges retenues contre eux) et des dates dans le cadre d'une étude du complot 19 août 1820 contre Louis XVIII. ©FARAUT (Vivien), <i>Les outils de représentation graphique de l'espace relationnel face au secret : le cas des conspirateurs du 19 août 1820</i> , Les Cahiers de Framespa. Nouveaux champs de l'histoire sociale, 2015, URL : http://journals.openedition.org/framespa/3233 (visité le 15/09/2020)	55
3.1	Modélisation conceptuel des données, des formats et des systèmes d'informations du projet Lectaurep ©L. Terriel, 2020, Diagrams.net	68

3.2	Un exemple de métadonnées EXIF extraites d'une image de répertoire numérisée et affichées en sortie d'un script Python (Cf. Annexes, Figure F.2) ©L. Terriel, 2020, <i>Pycharm</i>	70
3.3	Présentation du workflow pour le fichier pivot XML-TEI Lectaurep mis en place durant le stage ©L. Terriel, 2020, <i>Diagrams.net</i>	79
4.1	Document XML-TEI minimal conforme et valide ©L. Terriel, 2020, <i>Oxygen XML Editor</i>	82
4.2	Diagramme en oignon des différents niveaux de granularité des données Lectaurep pour représenter les imbrications dans le fichier pivot XML-TEI ©L. Terriel, 2020, <i>Diagrams.net</i>	84
4.3	Schéma minimal retenu pour l'exposition des données Lectaurep dans un fichier pivot XML-TEI ©L. Terriel, 2020, <i>Oxygen XML Editor</i>	87
4.4	Illustration des entrées et des sorties dans le teiHeader ©L. Terriel, 2020, <i>diagrams.net</i> (inspiré du document fourni par L. Romary, « The teiHeader at a glance »))	88
4.5	Structure proposée en TEI pour l'encodage des différentes granularités descriptives des éléments EAD <c> pour représenter un répertoire et ses contenus. ©L.TERRIEL, 2020, <i>Diagrams.net</i>	90
4.6	Imbrication des deux niveaux de <list> dans le <teiHeader> ©L. Terriel, 2020, <i>Diagrams.net</i>	92
4.7	Un exemple d'encodage d'informations EXIF dans une balise TEI <xenoData> ©L. Terriel, 2020, <i>Oxygen XML Editor</i>	94
4.8	Structure du fichier XML ALTO à gauche et du fichier XML TEI pivot et liens entre les éléments ©L. Terriel, 2020, <i>Diagrams.net</i>	97
4.9	Capture du terminal qui montre le bon déroulement du CLI Generator Lectaurep-TEI. ©L. Terriel, 2020, <i>Diagrams.net</i>	103
4.10	Illustration d'une transformation XSLT d'un document XML vers un document HTML ©L. Terriel, 2020, <i>Diagrams.net</i>	104
4.11	Exemple d'un fichier XML « catalogue » qui regroupe des fichiers XML EAD-EAC ©L. Terriel, 2020	106
5.1	Exemple de rapport fourni par kraken ©2015, <i>Kraken API</i> , URL : http://kraken.re/api.html	112
5.2	Capture d'écran du rapport produit par le prototype cerwer_tool.py ©L. Terriel, 2020, <i>Pycharm</i>	114
6.1	L'écosystème Python pour la science des données (<i>datascience</i>) ©F.Pennerath, Mineure « Data Science », Centrale Supélec	124

6.2	Environnement de transcription de Kraken utilisés pour créer des vérités terrains de tests ©L. Terriel, 2020, <i>Kraken</i>	129
6.3	Le concept de personne peut être représenté sous la forme d'un objet ©L. Terriel, 2020, <i>Diagrams.net</i>	131
6.4	Diagramme du module <code>SynSemTS.py</code> présentant les différentes classes ©L. Terriel, 2020, <i>Pycharm</i>	133
6.5	La page d'accueil : tableau de métriques ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i>	137
6.6	La page d'accueil : l'image et le graphique des opérations ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i>	138
6.7	La fonctionnalité <i>Show versus</i> ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i> . .	139
6.8	La fonctionnalité <i>Ranking errors</i> ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i> .	140
6.9	La fonctionnalité <i>Vizualize signals</i> ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i>	141
6.10	Exemples d'interprétations de la fonctionnalité <i>Vizualize signals</i> ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i>	142
6.11	La fenêtre « pop-up » qui permet de régler l'intervalle des séquences à visualiser dans le graphique de <i>Vizualize signals</i> ©L.TERRIEL, 2020, <i>Kraken-Benchmark</i>	142
6.12	Le Cas 2 représente un scénario utilisateur envisageable dans <i>Kraken-Benchmark</i> ©L. Terrie, 2020, <i>Diagrams.net</i>	145
7.1	<i>Set_material_defects</i> : les focus 1) et 2) correspondent à l'image <code>subject_1_robin_DAFANCH96</code> (N&B, étude XLVIII, notaire Jean-François Robin), présentant des tâches d'encre et des écritures marginales, des noirceurs et des forts contrastes ; le focus 3) <code>subject_2_rigault_FRAN_0187_16416_L-1.jpeg</code> , (couleurs, étude LXXXVI, notaire Jean-Paul Rigault), présentant un ruban adhésif sur la partie inférieure droite du document obstruant une partie des colonnes 6 et 7 ; le focus 4) <code>subject_3_michaux_DAFANCH96_MIC067000672-1.jpeg</code> , (N&B, étude VII, Pierre Michaux) présentant un ruban adhésif épais obstruant une partie du texte dans les colonnes 1, 2, 3, et 4 et une très mauvaise qualité de numérisation. ©AN-DMC, 2020	148

- 7.2 *Set_writing_defects* : le focus 1) correspond à l'image **subject_1_dufour_DAFANCH96_048MIC0** (N&B, étude XLVIII, notaire Jean Dufour), présentant une double flèche dans le coin inférieur gauche du répertoire au niveau de la colonne 1 ; le focus 2) correspond à l'image **subject_2_rigault_FRAN_0187_16428_L-1.jpeg**, (N&B, étude LXXXVI, notaire Jean-Paul Rigault), présentant un élément atypique situé au niveau du nombre « 25 » de la colonne 6 correspondant aux « dates » ; le focus 3) correspond à l'image **subject_3_dufour_DAFANCH96_048MIC08733_L-** (N&B, étude XLVIII, notaire Jean Dufour) présentant une calligraphie fracturée. ©AN/DMC, 2020 149
- 7.3 Derniers entraînements de modèles réalisés pour Lectaurep ©A. Chagué, 2020 154
- A.1 Exemple de minute notariale ©Archives nationales/DMC, Minute « Contrat de mariage entre Charles Camille Saint-Saëns, compositeur de musique, organiste de la Madeleine demeurant au 168 rue du Faubourg Saint-Honoré, et Marie-Laure Truffot, fille de Rodrigue Truffot, manufacturier au Cateau-Cambrésis », 18 janvier 1875, MC/ET/XXVI/1345 (cote originale), MC/RS//872, lien vers la SIV : https://www.siv.archives-nationales.culture.gouv.fr/siv/UD/FRAN_IR_041418/c1p6uqwjl1o3-x1v0cdl5wlgv(consulté le 14/09/2020).167
- A.2 Structuration en tableaux des répertoires ©BONHOMME (Marie-Laurence), *Défis et opportunités de la reconnaissance automatique d'écriture manuscrite pour les documents d'archives : l'exemple des répertoires des notaires de Paris*, Mémoire de recherche, École nationale des chartes, 2018, pp. 27. . 168
- A.3 Application web eScriptorium ©L. Terriel, 2020, eScriptorium 169
- A.4 Exemple du *Golden Set* et du *Random Set* stockés sur l'espace *Sharedocs* (Huma-num) ©L. Terriel, 2020, *Sharedocs* (Huma-num) 169
- A.5 Blog *hypotheses.org* Lectaurep ©L. Terriel, blog *hypotheses.org* Lectaurep . 170
- D.1 Schématisation du modèle de circulation des données dans *eScriptorium* souhaité par Lectaurep à terme ©L. Terriel, 2020, yEd 180
- D.2 Algorigramme du programme *Generator Lectaurep-TEI* pour simuler la visualisation d'un fichier XML-TEI pivot comprenant des données provenant de fichiers XML ALTO, EAD et EAC-CPF et des métadonnées EXIF provenant d'images. ©L. Terriel, 2020, Diagrams.net 181
- E.1 Fenêtre pour visualiser des détails concernant le modèle envoyé dans l'interface *Transkribus* ©A. Chague, 2020, *Transkribus* 184
- E.2 Fenêtre pour comparer la référence et la prédiction dans l'interface *Transkribus* ©A. Chague, 2020, *Transkribus* 184
- E.3 Algorigramme de *Kraken-Benchmark*. ©L. Terriel, 2020, Diagrams.net . . . 185

F.1	<i>Script Python pour effectuer de l'étiquetage morpho-syntaxique (POS) ©L.</i> Terriel, 2020	188
F.2	<i>Script Python pour afficher les métadonnées Exif d'une image ©L. Terriel,</i> 2020	189
F.3	<i>Script Python pour illustrer les différents scores de similarité syntaxique entre deux chaînes de caractères et leurs implémentations ©L. Terriel, 2020</i>	190

Lexique des termes informatiques

Liste non exhaustive

- **Algorithme** : Suite d'étapes réalisées par un programme informatique pour effectuer une tâche.
- **Back-office/Front-office** : Dans une application, désigne la partie visible par le client (*front-office*) et la partie qui concerne les systèmes d'informations (bases de données) et leur gestion, invisible pour l'utilisateur final (*back-office*).
- **Blog** : Type de site *web* qui permet la publication périodique d'articles scientifiques rendant compte de l'actualité d'un projet ou d'une thématique.
- **Interface en ligne de commande** : CLI ou *Command Line Interface* en anglais - interface homme machine dépourvue d'aspect graphique et où la communication s'effectue en mode texte, au moyen de lignes de commande (texte entré à l'aide du clavier) pour demander à l'ordinateur d'effectuer une opération.
- **Commit** : Dans un système de versionnage, commande qui permet de valider des modifications locales vers un référentiel central afin de les mettre à disposition.
- **Dépôt (informatique)** : *repository* en anglais - S'applique aux logiciels de gestion de versions, stockage organisé de données ; endroit où l'on dépose le code-source.
- **Docstring** : Chaîne de caractères pour documenter un segment spécifique du code informatique.
- **Fonction** : En programmation, « sous-programme » qui permet de réaliser des tâches répétitives pour alléger du code.
- **Interface graphique** : GUI ou *Graphical User Interface* en anglais - environnement qui permet l'interaction entre l'homme et la machine, à l'image de la métaphore du bureau dans la plupart des systèmes d'exploitation.
- **Interpréteur de commande** : Terminal informatique, logiciel compris initialement dans le système d'exploitation, il permet d'interpréter des commandes d'un utilisateur ou d'une utilisatrice dans un environnement dépourvu d'interface graphique.
- **Issue** : Dans une plate-forme de versionnage, peut s'apparenter à un billet qui permet d'émettre des suggestions ou qui fait état des bugs dans un contexte de

développement.

- **Merge** : Action de fusionner des branches (versions) différentes d'un dépôt informatique. (Cf. *pull request* (*Github*) ou *merge request* (*Gitlab*))
- **Module** : En développement, fichier contenant des fonctions, des classes ou des variables pouvant être importées dans un *script* pour en exploiter le contenu.
- **Package** - sym. *library* : Ensemble de modules de traitements spécifiques pouvant être importés.
- **Parser** : Programme informatique qui permet l'analyse syntaxique des éléments afin de leur donner une signification. (Cf. *parser XML*)
- **Push** : Dans un système de versionnage, commande qui après un *commit* permet d'envoyer les modifications d'un système local vers un référentiel central pour partager ces dernières.
- **Open-source** : Communauté et types de licences qui s'appliquent à des programmes ou à des logiciels, permettant la redistribution et la réutilisation de leurs code-sources.
- **Script** : Programme ou extrait de programme qui permet de réaliser une tâche prédéfinie (Cf. Algorithme).
- **Système d'exploitation** : *Operating System* (OS) en anglais - Ensemble de programmes qui permettent d'utiliser les ressources d'un ordinateur. Le logiciel système pilote les ressources matérielles de l'ordinateur et reçoit les instructions des usagers ou d'autres logiciels.
- **Logiciel de gestion de versions** : Logiciel qui permet de stocker des fichiers en conservant la chronologie de l'ensemble des modifications (versions) qui y ont été effectuées.
- **Tests unitaires** : En programmation, procédure qui permet de vérifier une partie précise d'un logiciel ou d'un programme pour s'assurer de son bon fonctionnement.

Table des matières

Résumé	v
Remerciements	vii
Liste des sigles et abréviations	ix
Introduction	3
I Le projet Lectaurep : un cas d'application de l'« intelligence artificielle » aux documents historiques	7
1 Lectaurep, un projet de recherche et développement en analyse et reconnaissance de document	9
1.1 Lectaurep, un enfant né de la rencontre du numérique et du patrimoine	9
1.1.1 La transformation digitale	9
1.1.2 Le mouvement de la numérisation des institutions patrimoniales dans les années 1990	10
1.1.3 L'essor des humanités numériques	11
1.1.4 Les potentialités des technologies OCR/HTR pour les institutions patrimoniales et leurs publics	12
1.1.5 La mise en place d'un projet HTR au département du minutier central des Archives nationales	14
1.2 La reconnaissance automatique des écritures pour le « plus grand minutier du monde »	15
1.2.1 Le département du minutier central et les répertoires de notaires . .	15
1.2.2 Le projet Lectaurep : cadre, avancées et objectifs de la phase 3 . . .	18
2 La reconnaissance automatique des écritures dans Lectaurep : un domaine de l'intelligence artificielle et du traitement automatique du langage naturel	23
2.1 Définir les composantes de l'intelligence artificielle dans le projet	23

2.1.1	Les champs de l'intelligence artificielle	23
2.1.2	L'étape de préparation et d'acquisition des données d'apprentissage de Lectaurep	26
2.1.3	Apprentissage et entraînement des modèles	34
2.1.4	La prédiction par la machine	35
2.2	Avant l'intelligence artificielle, la reconnaissance optique de caractères	38
2.2.1	Historique	38
2.2.2	Modèles de réseaux de neurones profonds appliquées à l'HTR	41
2.2.2.1	Historique des réseaux de neurones	41
2.2.2.2	Du neurone formel aux réseaux de neurones multi-couches : fonctionnement	42
2.2.2.3	Quel réseau de neurones pour l'HTR ?	44
2.3	Extraire, analyser et exploiter les données de l'HTR avec le traitement automatique du langage naturel	46
2.3.1	Les techniques du TAL	47
2.3.2	Applications et potentialités du TAL pour Lectaurep	50

II Représenter et homogénéiser dans un format pivot XML TEI les métadonnées Lectaurep 57

3 Enjeux et analyse des problématiques liées aux données Lectaurep et au format pivot XML-TEI	61
3.1 De l'importance de l'interopérabilité des données	61
3.1.1 Définitions et objectifs de Lectaurep	61
3.1.1.1 Rappels généraux : données et formats	61
3.1.1.2 La question de l'interopérabilité dans Lectaurep	65
3.1.1.3 Une solution, le format pivot XML	67
3.1.2 « Circonscrire un monde », un focus sur les données de Lectaurep	67
3.2 Le format pivot XML TEI, un choix réaliste ?	72
3.2.1 La TEI pour annoter les données et les métadonnées de Lectaurep	72
3.2.2 Faire converger les standards de données vers un fichier pivot XML-TEI : confronter des visions opposées sur le document	73
3.2.3 ...qui a pourtant fait ses preuves dans des projets de standardisation : les avantages spécifiques pour Lectaurep	76
4 Workflow et mise en place d'une première version du fichier pivot XML-TEI	81
4.1 Un canevas de travail : un fichier XML-TEI dans un environnement partagé et ouvert	81

4.2	Une première version du schéma pour le fichier pivot XML-TEI : repérage des données et choix d'encodage	83
4.2.1	Repérage des données et règles préalables à l'encodage	83
4.2.2	Encodage des éléments de description des répertoires (EAD et EAC) dans le <code>teiHeader</code>	88
4.2.3	Encodage des éléments des métadonnées EXIF dans le <code>teiHeader</code> via des éléments <code>xenoData</code>	93
4.2.4	Encodage des éléments ALTO correspondant aux zones et lignes de texte, et à la transcription dans l'élément <code>facsimile</code> et l'élément <code>body</code>	95
4.2.5	Compléments sur l'encodage du fichier pivot XML-TEI	98
4.3	Une ODD pour documenter, partager et valider le canevas XML-TEI	98
4.4	Les évolutions du fichier pivot XML-TEI pour la suite du projet	99
4.5	Simuler l'agrégation des données et la validation d'un fichier XML-TEI pivot avec un <i>script</i> Python	101
4.5.1	Les différentes étapes du programme : le script principal <code>main.py</code> .	102
4.5.2	Deux feuilles de styles XSLT pour obtenir des arborescences TEI .	104
4.5.3	Perspectives d'évolution pour le <i>Generator Lectaurep-TEI</i>	106
III	Développer une application en Python pour évaluer les modèles HTR	107
5	État de l'art pour l'évaluation des modèles de transcription entraînés avec le système HTR Kraken	111
5.1	Banc d'essai des outils existants : limites et avantages	111
5.2	Des métriques pour comparer la transcription automatique et la vérité terrain	115
5.2.1	La comparaison de chaînes de caractères	115
5.2.2	Estimer la similarité entre deux documents	119
5.2.3	Remarques complémentaires sur les métriques d'évaluation utilisées	121
6	Le développement d'une application : Kraken-Benchmark	123
6.1	Modélisation	123
6.1.1	Les objectifs fixés au début du développement	123
6.1.2	Un écosystème Python orienté pour la science des données et la conception d'application <i>web</i>	124
6.1.3	Les étapes de fonctionnement : retour sur quelques aspects de programmation	127
6.1.3.1	Un jeu de données pour réaliser des tests fonctionnels . . .	128
6.1.3.2	Deux niveaux d'interface pour un prototypage rapide . . .	129

6.1.3.3	La programmation orientée objet : une solution pour généraliser et mieux documenter le code	130
6.1.3.4	Retour sur les principales difficultés rencontrées	134
6.2	Suivi sur la conception et retour sur les usages de <i>Kraken-Benchmark</i>	135
6.2.1	La gestion du projet <i>Kraken-Benchmark</i>	135
6.2.2	Un tour d'horizon de l'interface <i>Kraken-Benchmark</i> et des fonctionnalités actuelles	137
6.3	Perspectives d'amélioration techniques pour l'application	143
7	Tests de <i>Kraken-Benchmark</i> sur les données Lectaurep	147
7.1	Préparation des jeux de données	147
7.1.1	Les images	147
7.1.2	Les modèles	150
7.1.3	Les vérités terrains	150
7.2	Méthodologie et résultats obtenus	151
7.3	Un bilan mitigé pour les tests dans <i>Kraken-Benchmark</i> ?	152
Conclusion		157
Annexes		163
A Sources et Ecosystème Lectaurep		165
A.1	histoire du projet lectaurep	165
A.2	Extraits du corpus des répertoires de notaires	165
A.3	Outils généraux utilisés dans Lectaurep	165
B Format pivot XML-TEI Lectaurep		171
C Application <i>Kraken Benchmark</i>		175
D Documents de travail pour le fichier pivot XML-TEI Lectaurep		179
E Documents de travail pour <i>Kraken-Benchmark</i>		183
F Scripts Python complémentaires		187
Bibliographie		195
Table des figures		221

233

Lexique des termes informatiques **227**

Table des matières **229**