ÉCOLE NATIONALE DES CHARTES
UNIVERSITÉ PARIS, SCIENCES & LETTRES

Kelly Christensen
licenciée ès enseignement musical
diplômée de master musicologie
diplômée de doctorat musicologie

Modélisation des transcriptions ALTO avec la TEI

En complétant le pipeline du projet *Gallic(orpor)a*

Mémoire pour le diplôme de master
« Technologies numériques appliquées à l'histoire »

2022

Résumé

Quand des modèles OCR (*Optical Character Recognition*) et HTR (*Handwriting Text Recognition*) extraient les données d'une ressource textuelle numérisée, les informations relatives à la structure physique de l'image risquent de se perdre. Un schéma XML standardisé qui s'appelle ALTO (*Analyzed Layout and Text Object*) a été créé afin de conserver et structurer ces données non-textuelles et géometriques en les tenant en relation avec le contenu textuel. La plupart des modèles OCR et HTR compte sur ce schéma. Cependant ALTO ne convient pas bien à l'édition numérique ni aux traitements automatique du langage. Les éditeurs et les chercheurs en lettres attendent un schéma XML plus courant dans le monde des humanités numériques : la TEI (*Text Encoding Initiative*). Il faut donc un mapping pour transformer un fichier ALTO en fichier TEI sans perdre aucune donnée lors du processus. Cette transformation automatisée permet à conserver les données particulières au schéma ALTO, telles que celles sur la segmentation et sur la structure physique du document numérisé, ainsi qu'à exploiter le contenu textuel de la ressource textuelle. La flexibilité de la TEI et son usage très répandu rendent le schéma idéal pour mieux valoriser les données produites par les modèles OCR et HTR.

Dans le cadre du stage pour obtenir le diplôme de Master 2 « Technologies numériques appliquées à l'histoire », ce mémoire porte sur la modélisation de la transformation de ALTO en TEI. Cette modélisation a été réalisée dans le cadre du projet *Gallic(orpor)a*, financé par la BnF (Bibliothèque nationale de France) lors d'un stage qui a eu lieu au sein du laboratoire ALMAnaCH (Automatic Language Modelling and Analysis & Computational Humanities) entre avril et juillet 2022.

Mots-clés : HTR, OCR, ALTO, TEI, TAL, édition numérique.

Informations bibliographiques : Kelly Christensen, *Modélisation des transcriptions ALTO avec la TEI. En complétant le pipeline du projet* Gallic(orpor)a, mémoire de master « Technologies numériques appliquées à l'histoire », dir. [Noms des directeurs], École nationale des chartes, 2022.

Remerciements

Mes remerciements vont tout d'abord à...

Introduction

Première partie

Présentation du projet

Chapitre 1

Qu'est-ce que l'HTR ?

1.1 L'objectif de l'HTR

Décrire les enjeux de la reconnaissance automatiques des caractères de texte sur une image. Donner des exemples de cette tâche avec une petite figure qui montre le processus.

FIGURE 1.1 – figure

1.2 L'histoire et l'évolution de la technologie

Décrire l'évolution de cette technologie, en commençant avec l'OCR.

1.3 Les deux approches actuelles

Expliquer qu'il y a actuellement deux approches : OCR et HTR.

1.3.1 L'OCR

Expliquer comment cette technologie plus ancienne compte sur les polices des caractères et a besoin des technologies de traitement automatique des langues.

1.3.2 L'HTR

Expliquer comment l'HTR réussit à s'entraîner sur les courbes des caractères écrits et peut donc compléter ses tâches sans besoin d'un modèle TAL derrière. (Au moins, c'est actuellement ce que j'ai compris des explications d'Ariane Pinche) Justifie pourquoi l'HTR a été privilégié dans le projet Gallic(orpor)a, même pour les imprimés.

Chapitre 2

Au commencement, il y avait les *guidelines SegmOnto*

2.1 La problématique

...

2.2 Les solutions proposées

Plusieurs projets avaient proposé des solutions au manque de cohérence dans la manière par laquelle la communauté scientifique caractérise les composants d'un texte numérisé. Hors de la France, les vocabulaires codicologiques ont été élaborés notamment en anglais pour les manuscrits médiévaux. Un exemple important est la base de données DigiPal (Digital Resource and Database of Palaeography, Manuscripts and Diplomatic), qui n'est plus mis à jour mais qui a été développé au sein du département des humanités numériques à King's College London. [**noauthor_digipal_2011**] L'un de ses auteurs, Peter Stokes, travaille actuellement en France et continue dans la même veine en collaborant avec des chercheurs français sur le projet *SegmOnto*. [**gabay_segmonto_2021**] Avant la création des *guidelines* de *SegmOnto*, le développement des vocabulaires codicologiques en France était toujours basé sur le modèle de Denis Muzerelle et son *Vocabulaire codicologique*, que nous expliquons prochainement.

2.2.1 Le Vocabulaire international de la codicologie

En 1985, Denis Muzerelle a conçu un vocabulaire codicologique qui avait pour but de fournir des médiévistes avec des termes uniformes pouvant décrire les aspects d'un manuscrit. [**muzerelle_vocabulaire_1985**] Depuis l'apparition de son vocabulaire en français, des autres chercheurs sont venus pour adapter les termes de Muzerelle en d'autres langues. Marilena Maniaci a publié une version du *Vocabulaire codicologique* pour l'italien

en 1996. [**maniaci_terminologia_1996**] Pilar Ostos, Luisa Pardo, et Elena Rodríguez en ont créé un pour l'espagnol l'année suivante. [**ostos_vocabulario_1997**] Parfois appelé le *Vocabulaire international de la codicologie*, l'édition multilingue du *Vocabulaire codicologique* que Muzerelle a commencé en 1985 était maintenue jusqu'à l'édition d'une version 1.1. en 2002-2003. (besoin de citation)

2.2.2 La Codicologia

Aujourd'hui, la paléographie et l'étude des manuscrits peuvent profiter de l'application web *Codicologia* qui réunit le *Vocabulaire codicologique* ainsi que deux autres bases de données similaires : le projet multilingue *Lexicon* et le *Glossaire codicologique arabe*. Ses trois bases de données spécialisent dans divers écritures. Comme précisé avant, le *Vocabulaire codicologique* fournit une liste de termes en français, italien, espagnol, et anglais. Piloté par Philippe Bobichon, le projet *Lexicon* présente un vocabulaire en français pour décrire les manuscrits écrits en latin, roman, grec, hébreu, et arabe. [**bobichon_lexicon_2009**] Un vocabulaire spécialisé plus profondément pour l'arabe est élaboré dans le *Glossaire codicologique arabe* d'Anne-Marie Eddé et Marc Geoffroy. [**noauthor_glossaire_2002**] Ce dernier a été conçu au sein de l'Institut de recherche et d'histoire des textes après les modèles de Muzerelle et le vocabulaire codicologique en arabe d'Adam Gacek. [**gacek_arabic_2001**]

L'application web *Codicologia* rassemblent ces projets et présente un vocabulaire bien étendu. Par exemple, *Codicologia* fournit 15 termes pour décrire une faute d'écriture dans un manuscrit. Certains de ses termes possèdent eux-même plusieurs définitions que les divers bases de données fournissent. Le terme *caviarder*, par exemple, a une définition courte dans le vocabulaire français de Muzerelle.

> Supprimer un mot, un passage..., en le recouvrant largement d'encre, de façon à ce qu'il ne puisse être lu. [**muzerelle_caviarder_2011**]

Selon le `Lexicon` de Bibichon, par contre, le *caviarder* se définit d'une manière plus détaillé et vise à expliquer l'étymologie du mot afin de préciser son usage dans le cadre des manuscrits des divers écritures.

> Le mot apparaît en 1907 (noircir à l'encre) : il désigne alors un procédé appliqué par la censure russe, sous Nicolas Ier. Dans certains manuscrits grecs, le détail rempli d'encre est surmonté d'un point et d'un trait court destinés à le neutraliser. Ce procédé est très souvent utilisé parmi d'autres, pour la censure des manuscrits hébreux effectué sous l'autorité de l'inquisition, en Italie, à la fin du xvie siècle et au début du xviie. [**bobichon_caviarder_2011**]

Étant élaboré à partir d'un corpus très diversifié, le *Lexicon* de Bibichon a moins de termes qu'a le *Vocabulaire codicologique* mais ses termes sont plus généralisés. Le vocabulaire de Muzerelle, par contre, fait plus de distinctions entre les aspects d'un manuscrit et

donc a plus de termes distincts par rapport aux deux autres vocabulaires de l'application *Codicologia*.

En réunissant les trois bases de données, sans privilégier aucun, *Codicologia* présente un vocabulaire codicologique vraiment vaste. Cependant, l'application *Codicologia*, comme toutes ses bases de données, vise à répondre au manque de cohérence dans la manière par laquelle la communauté scientifique décrit les manuscrits. Le grandeur de son vocabulaire pose un problème à cet objectif. Ayant plus de deux milles termes en français—certains d'entre eux ont eux-même plusieurs définitions—la solution proposée par *Codicologia* livre un vocabulaire bien harmonisé et documenté mais trop étendu pour être appliqué à l'échelle dans une approche informatique. Sans un corpus d'entraînement gigantesque, qui couterait une somme énorme, l'apprentissage automatique ne peut pas faire de distinction au niveau des termes conçus par Muzerelle et les autres auteurs des bases de données de *Codicologia*. Aujourd'hui, un modèle ne peut pas s'entraîner sur des milles des étiquettes possibles et arriver à distinguer entre, par exemple, 15 types de faute d'écriture. Un humaine peut le faire, et pour cette raison les bases de données de *Codicologia* sont utiles. Mais leurs vocabulaires ne conviennent pas bien à une approche informatique.

2.3 Les *guidelines* de *SegmOnto*

Le projet *SegmOnto* propose un vocabulaire plus petit et pourtant peut décrire une grande diversité de documents historiques, y compris les manuscrits et les imprimés. Cet objectif est encore plus compliqué à achever qu'un vocabulaire spécialisé aux manuscrits. Décrire les documents d'une diachronie longue, et sans préférence d'une écriture en particulier, exige un équilibre délicat entre la généralité et la particularité. Les *guidelines* du projet *SegmOnto* essaient de limiter le nombre de termes dans son vocabulaire sans priver un terme d'une identité distincte. Les *guidelines* se divisent en deux catégories génériques : les régions ou "zones" d'une page et les lignes de texte ou d'écriture sur la page. Chaque catégorie se compose d'une liste des étiquettes, chacune de lesquels cherche à parvenir à l'équilibre. Une étiquette devrait pouvoir être appliquée à soit un manuscrit, soit un imprimé, de peu importe quelle langue et quelle écriture.

2.3.1 Les zones

— CustomZone
— DamageZone
— DecorationZone
— DigitizationArtefact
— DropCapitalZone

- MainZone
- MarginTextZone
- MusicZone
- NumberingZone
- QuireMarksZone
- RunningTitleZone
- SealZone
- StampZone
- TableZone
- TitlePageZone

2.3.2 Les lignes

- CustomList
- DefaultLine
- DropCapitalLine
- HeadingLine
- InterlinearLine
- MusicLine

Chapitre 3

Le rêve du projet *Gallic(orpor)a*

3.1 Le contexte du projet

Présenter les institutions qui soutiennent le projet (École nationale des chartes, Inria, Université de Genève, DataLab de la BnF) et les projets qui l'ont précédé et sur lesquels *Gallic(orpor)a* compte.

3.2 L'objectif du projet

Le projet a pour but d'arriver des images numérisées à un document numérique sans besoin d'un éditeur. L'idée est qu'avec le produit du pipeline, qui sera un texte prédit et pré-éditorialisé, un individu qui n'est pas forcement spécialisé dans l'informatique peut facilement le prendre et effectuer des analyses et/ou éditer le texte du document dont les images de ses pages ont été traités.

3.3 Le pipeline

Présenter le pipeline. Réutiliser l'information qu'on a générée pour le présenter à la BnF.

Deuxième partie

Exposition de la préparation et du travail d'analyse

Chapitre 4

Un pipeline visant à tout rassembler

4.1 Les données d'entraînement HTR

J'ai généré des données d'entraînement sur eScriptorium en suivant les mêmes conseils donnés aux vacataires chargés à segmenter et transcrire les images d'un document selon les normes de *SegmOnto*. Je me suis familiarisée avec leur travail et la création du corpus d'or qui serviraient à l'entraînement du modèle HTR. J'ai aussi surveillé la discussion entre les vacataires et les chefs du projet, afin de poser et de répondre aux questions généralisées. Grâce à ces tâches, je comprends mieux le défi d'harmoniser la création d'un corpus d'or par en groupe.

4.2 Les données d'entraînement TAL

Certains vacataires ont travaillé sur les textes extraits des transcriptions que les autres vacataires ont faites sur eScriptorium, et ils ont créé des données textuelles qui serviraient à l'entraînement des modèles TAL. Afin de les aider dans la création de ce deuxième corpus d'or, j'ai créé un workflow automatisé sur les dépôts GitHub du projet qui extrait automatiquement les lignes de texte des fichiers ALTO et divise le texte en segments selon les signes de ponctuation. La création d'un workflow sur GitHub m'ai appris comment le faire plus tard pour l'application `alto2tei`.

4.3 La fin du pipeline : TEI Publisher

J'ai assisté à un atelier sur le logiciel TEI Publisher afin de mieux comprendre les objectifs downstream de l'application `alto2tei`.

Chapitre 5

L'analyse des structures des données XML

5.1 XML-ALTO

5.1.1 Qu'est-ce qu'est le format ALTO ?

Décrire la création, le suivi, et l'objectif du format XML ALTO : enregistrer les infos sur la structure d'une image segmentée.

5.1.2 La structure des fichiers XML-ALTO

Montrer la structure des données d'un fichier ALTO dans les deux formats qui sortent de Kraken : (1) ligne de texte encodé dans la balise `<TextLine>`, qui sort de Kraken via l'interface d'eScriptorium, et (2) ligne de texte encodé au niveau du glyph, qui sort directement de la ligne de commande de Kraken.

5.2 XMl-TEI

5.2.1 Qu'est-ce qu'est la TEI ?

Décrire la création, le suivi, et l'objectif de la TEI.

5.2.2 Les éléments de base de la TEI

Expliquer qu'il y a deux éléments essentiels de la racine, le `<teiHeader>` et le `<body>`. Ensuite expliquer l'utilité de l'élément facultatif `<sourceDoc>` et expliquer pourquoi il convient bien aux données de structure d'un fichier ALTO.

Chapitre 6

À la recherche des métadonnées

6.1 Uniquement l'essentiel

6.1.1 Documents de divers types et de plusieurs époques

Expliquer le défi de modéliser un `<teiHeader>` qui est à la fois assez généralisé pour convenir aux divers documents et assez précisé pour servir aux utilisateurs et à la recherche.

6.1.2 Exemples des métadonnées souhaitées

Donner un exemple des métadonnées d'un imprimé (cf. fig. 6.1) :

```
<sourceDesc>
  <biblStruct>
    <monogr>
      <author xml:id="author">
        <persName>
          <surname>Balzac</surname> <!-- auteur -->
          <forename>Honoré</forename>
        </persName>
      </author>
      <title>The Wild Ass's Skin</title> <!-- titre -->
      <editor role="translator">Ellen Marriage</editor>
      <editor role="preface">George Saintsbury</editor>
      <pubPlace key="FR">Paris</pubPlace>
      <imprint>
        <pubPlace>London</pubPlace> <!-- lieu de publication -->
        <publisher>Dent</publisher> <!-- éditeur -->
        <date when="1906">1906</date> <!-- date de publication -->
      </imprint>
    </mongr>
  </biblStruct>
```

FIGURE 6.1 – Exemple des métadonnées d'un imprimé encodées en TEI (emprunté de `teibyexample.org` – à changer)

Et donner un exemple d'un incunable (cf. fig. 6.2) :

```
<sourceDesc>
  <msDesc>
    <msContents>
      <biblStruct>
        <monogr>
          <author xml:id="author">
            <persName>
              <surname>Tory</surname> <!-- auteur -->
              <forename>Geoffroy</forename>
            <persName>
          </author>
          <title>Champ fleury</title> <!-- titre -->
          <imprint>
            <pubPlace>Paris</pubPlace> <!-- lieu de publication / lieu d'
    apparition -->
            <publisher>
              <persName>
                <surname>Gourmont</surname> <!-- éditeur -->
                <forename>Gilles de</forename>
              </persName>
            </publisher>
          </imprint>
        </mongr>
      </biblStruct>
```

FIGURE 6.2 – Exemple des métadonnées d'un incunable encodées en TEI (emprunté du cours TEI de J-B Camps 2015 – à changer)

Expliquer comment l'objectif du projet *Gallic(orpor)a* de traiter des documents d'une diachronie longue pose un défi à la récupération et encodage généralisée des métadonnées.

6.2 Où se trouvent les métadonnées des sources de Gallica

Présenter les deux sources de métadonnées ciblées par l'application `alto2tei`.

6.2.1 L'IIIF Image API

L'API du manifest IIIF contient des données rudimentaires sur le document. Elles sont envoyées dans un format JSON. Donner un exemple (cf. fig. 6.3) :

6.2.2 L'API SRU de la BnF

L'API du catalogue général de la BnF contient des données bien précises sur le document. Elles sont envoyées dans un format XML-Unimarc. Donner un exemple (cf. fig. 6.4).

```
{"Metadata":
    {
    "Label": "Title",
    "Value": "The Wild Ass's Skin", # titre

    "Label": "Creator",
    "Value": "Honoré Balzac" # auteur
    }
}
```

FIGURE 6.3 – Exemple des métadonnées envoyées par l'API IIIF

```
  <mxc:datafield tag="200" ind1="1" ind2=" ">
    <mxc:subfield code="a">The Wild Ass's Skin</mxc:subfield> <!-- titre --
    >
    <mxc:subfield code="b">Texte imprimé</mxc:subfield>
  </mxc:datafield>
  <mxc:datafield tag="210" ind1=" " ind2=" ">
    <mxc:subfield code="a">Londres</mxc:subfield> <!-- lieu de publication
    -->
    <mxc:subfield code="c">Dent</mxc:subfield> <!-- éditeur -->
    <mxc:subfield code="d">1906</mxc:subfield> <!-- date de publication -->
  </mxc:datafield>
[...]
  <mxc:datafield tag="700" ind1=" " ind2="|">
    <mxc:subfield code="a">Balzac</mxc:subfield> <!-- auteur -->
    <mxc:subfield code="b">Honoré</mxc:subfield>
  </mxc:datafield>
```

FIGURE 6.4 – Exemple des métadonnées envoyées par l'API SRU de la BnF

6.3 Une solution

Présenter les métadonnées du document qu'on a déterminé d'être essentielle / assez généralisées parmi les divers types de document.

Troisième partie

Mise en opérationnelle du projet

Chapitre 7

La génération du `<teiHeader>`

7.1 La récupération des données

Parler de la récupération des données depuis les deux APIs discutés (cf. fig. 7.1) ainsi que le fichier YAML de configuration.

7.1.1 Du manifest IIIF à un dictionnaire Python

Montrer le mapping des données du manifest au dictionnaire Python.

7.1.2 De l'Unimarc à un dictionnaire Python

Montrer le mapping des données du catalogue au dictionnaire Python.

7.2 L'analyse des données

Parler de la stratégie d'atténuation des risques en sélectionnant les données fiables. Les métadonnées du catalogue général de la BnF sont utilisées uniquement si le même exemplaire physique du document numérisé sur Gallica a bien été trouvé. Sinon, on risque de mettre les données d'un autre exemplaire de l'oeuvre que celui qui a été transcrit et donc introduire des fausses données, tel que le cote ou même l'éditeur et la date de publication de l'exemplaire.

7.3 Le modèle du `<teiHeader>`

Montrer le mapping des données au `<teiHeader>`.

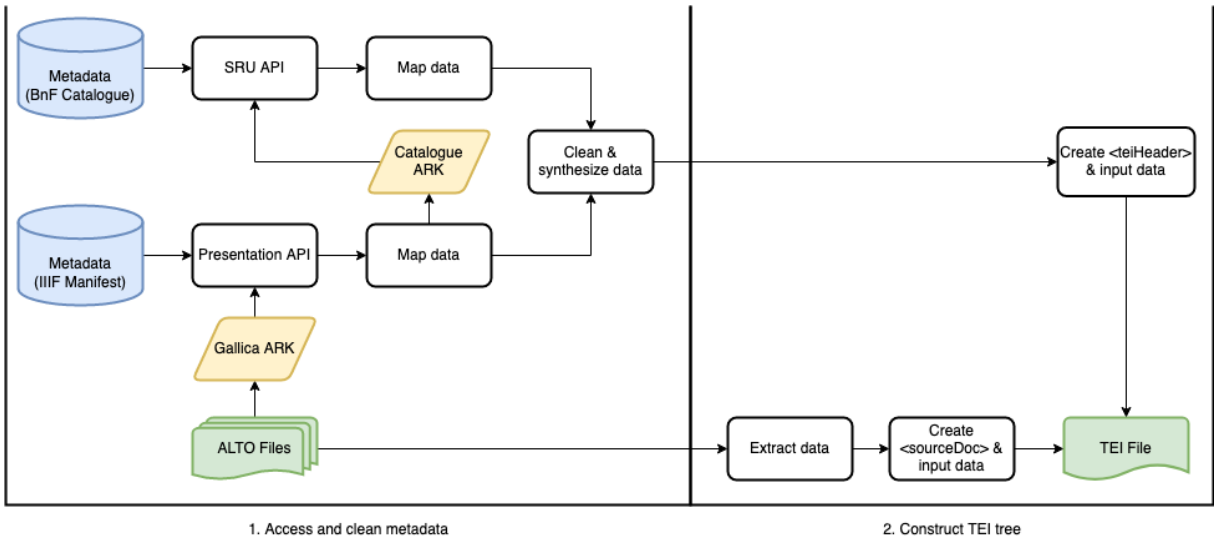

FIGURE 7.1 – Workflow

Chapitre 8

La modélisation de la <sourceDoc>

8.1 Le modèle du <sourceDoc>

Montrer le mapping des données prédites par les modèles HTR et segmentation vers les éléments TEI de la <sourceDoc>.

8.1.1 Niveau de la ligne de texte

Des exemples / tables ...

8.1.2 Niveau d'un mot ou d'une espace

Des exemples / tables ...

8.1.3 Niveau du glyphe

Des exemples / tables ...

8.2 Documenter ce modèle dans l'ODD

Présenter le travail d'avoir écrit l'ODD *SegmOnto-Gallicorpora*.

Chapitre 9

Les données textuelles produites

9.1 La génération du `<body>` grâce au lexique *SegmOnto*

Expliquer comment j'ai généré le `<body>` en prenant les lignes de texte de la `<sourceDoc>` selon leurs étiquettes.

9.2 L'analyse linguistique

Parler d'un travail futur qui pourra prendre le fichier XML-TEI que j'ai créé et extraire les lignes de texte du `<body>` et les passer aux modèles TAL pour faire des analyses linguistiques. Montrer un exemple de ces données et comment cela marcherait avec un travail préliminaire que j'aurai fait d'ici la fin de stage et/ou le travail d'un ancien stagiaire que j'ai appliquée aux données Gallic(orpor)a.

Conclusion

Annexe A

Données

Données du projet.

Table des figures

Liste des tableaux

Table des matières