



Étude sur l'enrichissement et le balisage automatique des débats
parlementaires

Master Humanités numériques



Présenté par :

Brunel TCHEKELI

Sous la direction de :

M. Jean-Philippe MAGUE

Tuteur et Tutrice de Stage :

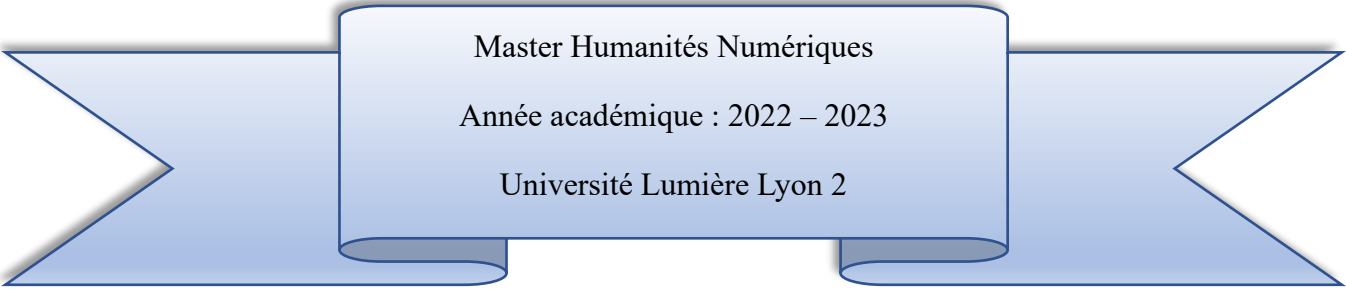
M. Pierre VERNUS

Mme Marie PUREN

Table des matières

Remerciements	3
Introduction	4
1. État de l'art : rapide histoire des débats parlementaires et leur importance pour l'histoire contemporaine	7
1.1. Le parlementarisme en France : Contexte historique et importance des Débats.....	7
1.2 L'avènement de la publicité formelle des discussions parlementaires	8
1.3. L'invention du compte rendu intégral des débats en France	8
1.4. Conserver l'histoire parlementaire	10
1.5. Numérisation : Modernisation et Accessibilité.....	11
1.6. Au-delà de la numérisation : Repenser l'interaction et l'éditorialisation des archives numériques.....	12
2. Fondements, inspiration et présentation du projet AGODA	14
2.1. Fondements et inspirations du projet.....	14
2.2. Présentation du projet AGODA	15
2.3. Objectifs principaux	16
2.4. La chaîne de traitement du projet (Workflow)	16
2.5. Porteurs du projet.....	19
2.6. Les projets similaires et les avancées dans le domaine	21
Problématique	23
3. Mise en œuvre de l'encodage : les enjeux du balisage automatique.	25
3.1 La conversion des fichiers : De la notation JSON à l'encodage XML-TEI.....	25
3.2 L'adoption du XML-TEI pour le projet AGODA.....	27
3.3. L'Adoption de l'ODD.....	29
3.4. Rappel sur l'objectif du stage et travail réalisé.....	30
3.5. Choix du langage et des bibliothèques pour le traitement automatique	31
3.6. A propos du script mis en place	32
3.7. Balisage physique	33
3.8. Balisage sémantique	46

3.9. Vérification de la conformité des fichiers TEI au schéma utilisé.....	51
4. Résultats et réflexion	52
4.1 Présentation des résultats obtenus lors du stage	52
4.2 Analyse des avantages et des limites de la méthode employée	52
4.3 Réflexion sur l'apport du travail réalisé.....	54
4.4. Retour sur l'expérience personnelle du stage.....	55
5. Perspectives.....	57
5.1 Pistes d'amélioration de la chaîne de traitement.....	57
5.2 Recommandations pour l'automatisation des tâches avec des algorithmes entraînés	58
Conclusion.....	59
Bibliographie.....	60
Annexes	65



Master Humanités Numériques
Année académique : 2022 – 2023
Université Lumière Lyon 2

Remerciements

Je souhaite adresser mes vifs remerciements aux personnes qui m'ont aidé dans la réalisation de ce mémoire et qui m'ont apporté un soutien méthodologique.

Ainsi, j'adresse mes remerciements à mes tuteurs de stage, Madame Marie PUREN et Monsieur Pierre VERNUS pour leur encadrement, leur disponibilité, et leur bienveillance durant ce stage.

Je remercie également mon tuteur académique, Monsieur Jean-Philippe MAGUE pour son accompagnement, ses conseils et sa disponibilité tout au long de ce travail.

Je tiens également à remercier Madame Morgane PICA du LARHRA qui a bien voulu m'apporter ses conseils et quelques documentations quand je bloquais sur mes lignes de codes.

Enfin, un grand merci à tous ceux qui ont participé à la relecture et à la reformulation de certaines phrases de ce mémoire.

Introduction

La France, avec sa riche histoire parlementaire, a toujours accordé une importance primordiale à la conservation de ses débats et documents parlementaires. Ces archives offrent une fenêtre sur l'évolution politique, sociale et économique du pays. La numérisation massive des archives historiques ouvre de nouvelles perspectives à la recherche en histoire et en sciences humaines. Parmi ces archives, les débats parlementaires constituent une source inestimable pour comprendre l'histoire politique, sociale, économique et culturelle de la période contemporaine, ainsi que pour approfondir l'histoire du droit. Toutefois, leur exploitation est souvent entravée par la disponibilité limitée de ces corpus, qui ne sont parfois accessibles qu'en tant qu'images ou textes bruts.

C'est dans ce contexte que s'inscrit le projet AGODA (Analyse sémantique et Graphes relationnels pour l'Ouverture et l'étude des Débats à l'Assemblée nationale). Fruit d'une collaboration entre le Laboratoire de recherche historique Rhône-Alpes (LARHRA) (CNRS-Lyon 2), le laboratoire Méthodes Numériques en Sciences Humaines et Sociales (MNSHS) à EPITA Paris, et ALMAnaCH (Automatic Language Modelling and Analysis & Computational Humanities) à Inria Paris, AGODA a pour ambition de créer une plateforme novatrice de consultation et d'exploration des débats parlementaires retranscrits dans le Journal officiel de la République française¹.

Le projet AGODA bénéficie du soutien du DataLab de la Bibliothèque nationale de France pour l'année 2021-2023, qui finance cinq projets pilotes, dont AGODA, afin de favoriser l'exploitation optimale des corpus numérisés disponibles dans ses archives. L'objectif premier du projet AGODA consiste à éditer et enrichir sémantiquement un vaste corpus de débats parlementaires, plus précisément ceux qui se sont tenus entre 1889 et 1893 à la Chambre des députés, disponibles sur Gallica.

La revalorisation des comptes rendus in extenso des débats parlementaires, un patrimoine méconnu et sous-utilisé, représente un enjeu majeur pour le projet AGODA. L'objectif central est de créer une plateforme en ligne de consultation et d'exploration des débats parlementaires

¹ Journal Officiel de La République Française - 70 Années Disponibles, Gallica

de la Chambre des députés de la III^e République, à partir des numérisations disponibles sur Gallica du Journal officiel de la République française.

Pour atteindre cet objectif, le projet AGODA réunit historiens et informaticiens, qui travaillent de concert pour faciliter l'accès et l'exploitation de ces documents à la fois pour les chercheurs et le grand public grâce aux technologies numériques. Cette démarche s'inscrit dans une volonté de repenser la circulation du savoir en donnant aux internautes de nouveaux moyens pour accéder à ce patrimoine historique, le comprendre et l'explorer.

Pour parvenir à exploiter efficacement cette source riche et abondante, le projet suit une méthodologie de traitement bien définie et accorde une grande importance à l'automatisation des tâches (automatisation future des commentaires lors de l'océrisation ; automatisation actuelle du processus d'encodage (entendu ici dans le sens de balisage) etc.). L'énorme quantité de données générées (10418 images numérisées à océriser) nécessite des méthodes automatiques pour assurer une gestion efficace. L'encodage automatique répond à ce besoin en transformant les données en formats exploitables tels que XML ou HTML.

La chaîne de traitement mise en place est essentielle pour rendre les données textuelles manipulables et exploitables. Elle débute par l'océrisation, c'est-à-dire la reconnaissance optique de caractères, pour transformer les numérisations en textes éditables.

En vue de la réalisation de cette plateforme de consultation, AGODA a choisi d'utiliser l'application TEI Publisher, reposant sur une base de données XML (eXist-db). La TEI (Text Encoding Initiative) est un standard international pour la représentation de textes numériques dans les domaines des sciences humaines et sociales, permettant de structurer et d'annoter de manière sémantique les documents. Cette approche permettra une exploration fine des contenus, offrant ainsi aux historiens et chercheurs de diverses disciplines (droit, sciences politiques, sociologie, linguistique, etc.) un accès privilégié à cette source historique de premier plan.

Le présent stage s'inscrit dans la continuité des travaux déjà entrepris par Fanny Lebreton lors de son stage de M2 en 2022 au sein du projet AGODA². Ce dernier a déjà permis d'établir les fondations nécessaires à la mise en place de la chaîne de traitement des documents, ainsi que d'élaborer un schéma XML pour l'annotation structurée et sémantique des débats parlementaires. Le stage proposé constitue un maillon essentiel du projet AGODA en contribuant activement à la mise en place d'une chaîne de traitement permettant l'édition et l'enrichissement sémantique d'une partie restreinte de ce corpus historique, à savoir les débats parlementaires de la période 1889-1893.

² <https://github.com/FannyLbr>

Parmi les tâches qui seront accomplies lors du stage, on retrouve la préparation des textes en vue de leur publication, avec des post-corrections et des annotations en TEI. Le processus d'automatisation du balisage des fichiers textes issus de l'océrisation sera également amélioré, en ajustant et complétant des scripts Python. Une démarche de vérification automatique de la conformité des fichiers TEI au schéma utilisé sera implémentée pour garantir la qualité des données.

Ce mémoire se focalise sur l'automatisation du balisage des fichiers et la procédure de vérification automatique de leur conformité aux normes TEI. Tout d'abord, nous présenterons le projet AGODA, en y associant un état de l'art qui mettra en lumière les fondements du projet et les démarches semblables déjà en cours. Ensuite, nous explorerons en profondeur la technique de balisage, en l'éclairant par divers exemples. Enfin, nous porterons un regard critique sur la méthodologie utilisée, terminant notre analyse par des conseils et des propositions pour perfectionner la phase de balisage automatique.

1. État de l'art : rapide histoire des débats parlementaires et leur importance pour l'histoire contemporaine

1.1. Le parlementarisme en France : Contexte historique et importance des Débats

La France, pays doté d'une riche histoire politique et culturelle, a traversé de nombreuses époques marquantes, des monarchies aux républiques, en passant par les empires. Au cœur de ces transformations, le parlementarisme s'est imposé comme un pilier essentiel de la démocratie française. Les débats parlementaires, reflets des préoccupations, des tensions et des aspirations de la nation, constituent une source inestimable pour comprendre l'évolution de la société française. Ils offrent une fenêtre sur les décisions politiques, les débats idéologiques et les mouvements sociaux qui ont façonné le pays.

Les origines du parlementarisme en France remontent au Moyen Âge avec la création des premières assemblées, telles que les États généraux. Bien qu'initialement conçues comme des organes consultatifs pour le roi, ces assemblées ont jeté les bases des futurs débats parlementaires en offrant un espace pour la discussion et la négociation entre les différentes classes de la société.

La révolution française de 1789 a marqué un tournant décisif dans l'histoire du parlementarisme français. L'abolition de la monarchie et la proclamation de la République ont donné naissance à de nouvelles institutions, notamment l'Assemblée nationale, où les débats sur la forme et la nature du gouvernement ont été particulièrement intenses. Ces discussions ont non seulement défini la trajectoire politique de la France, mais ont également établi le rôle central du parlementarisme dans la gouvernance du pays.

Au XXe siècle, la France a connu plusieurs régimes politiques, de la Troisième République à la Cinquième République actuelle. Chaque changement de régime a été accompagné de débats parlementaires cruciaux sur des questions telles que les droits de l'homme, la laïcité, la décolonisation et l'intégration européenne. Ces débats ont non seulement influencé la politique intérieure, mais ont également eu un impact sur la position de la France sur la scène internationale.

1.2 L'avènement de la publicité formelle des discussions parlementaires

En France, le principe de la publicité des débats parlementaires a été reconnu de facto depuis l'été 1789 et ensuite en droit avec la constitution du 3 septembre 1791. Ce principe stipulait que les délibérations du Corps législatif seraient publiques et que les procès-verbaux de ses séances seraient imprimés (Gaudillère, 2008).

Le concept de publicité prend une forme particulière dans le contexte des assemblées politiques et judiciaires, ainsi que dans la relation avec les médias. Deux types de publicité sont mis en avant : la publicité matérielle et la publicité par la presse (Lavoinne, 2022). Ces deux modes de constitution du public et de rapport au discours sont analysés à travers l'histoire, depuis le XVIIIe siècle jusqu'à l'époque contemporaine.

La publicité matérielle implique la présence physique du public lors des séances d'un tribunal ou d'une assemblée. Cependant, ce public est soumis à des normes de comportement strictes pour garantir la sérénité des débats. Cette publicité matérielle crée un public passif, présent en tant qu'assistance mais non destinataire des discours. Les régulations du comportement dans les tribunes et la nécessité de maintenir le calme dans l'enceinte renforcent cette idée d'un public spectateur.

D'un autre côté, la publicité par la presse repose sur le fait que le public lisant peut émettre des jugements sur les discours. Les textes imprimés sont destinés à être lus dans l'intimité, permettant ainsi une réflexion froide et rationnelle. Cette idée de lecture détachée des passions collectives est vue comme un facteur de progrès et d'égalité politique.

Le XIXe siècle a été marqué par une lutte pour l'application de ce principe aux différentes assemblées politiques délibérantes, avec un débat entre ceux qui considéraient que la publicité matérielle suffisait et ceux qui insistaient sur l'importance de la publicité imprimée, notamment à travers les journaux et leurs comptes rendus (Lavoinne, 2022).

L'étude de cette relation entre la publicité matérielle et symbolique des débats met en évidence un aspect crucial : La publicité matérielle était une condition nécessaire, mais pas suffisante, pour garantir la transparence et la responsabilité des assemblées politiques. La publicité symbolique, réalisée par le biais d'imprimés tels que les journaux, était considérée comme un moyen de donner une portée plus large aux débats, de les rendre accessibles à un public plus large et de les enraciner dans la mémoire collective (Morel, 2018).

1.3. L'invention du compte rendu intégral des débats en France

Pendant la Révolution française, les comptes rendus de séance souffraient de deux problèmes majeurs. Tout d'abord, ils manquaient de fiabilité en raison de l'orientation politique des journaux qui les publiaient (Coniez, 2010). Les comptes rendus étaient souvent favorables à

certaines camps politiques et déformaient les interventions des orateurs adverses. Ensuite, les comptes rendus de la Révolution étaient des résumés analytiques, c'est-à-dire des synthèses plus ou moins détaillées en fonction de l'intérêt supposé du public pour les débats en cours.

En 1791, le Journal Logographique tenta de concurrencer le Moniteur en proposant un compte rendu intégral des débats. Cependant, ce projet échoua en raison des difficultés techniques et de l'épuisement des rédacteurs. D'autres tentatives de sténographie se développèrent sous la Révolution, mais elles ne parvinrent pas à réaliser un compte rendu intégral des séances.

Ce n'est qu'au cours des années 1800 que de véritables systèmes sténographiques se développèrent en France, inspirés des travaux du pionnier britannique Samuel Taylor et adaptés au français par Théodore-Pierre Bertin. Jean-Baptiste Breton et Hippolyte Prévost, disciples de Bertin, jouèrent un rôle clé dans cette évolution (Coniez, 2010).

Lagache et Prévost, conscients des défis liés à la prise de notes sténographiques lors de séances longues et agitées, organisèrent un roulement rapide des rédacteurs en séance. Ils introduisirent également des réviseurs, des sténographes plus expérimentés chargés de corriger et de réviser le travail des rédacteurs en temps réel. Cette approche permettait d'obtenir un compte rendu fidèle et exhaustif des débats parlementaires.

En parallèle, Lagache et Prévost développèrent une approche rédactionnelle du compte rendu intégral en privilégiant la clarté et la compréhension du discours. Ils ne se contentaient pas de transcrire littéralement les propos des orateurs, mais les remaniaient pour les rendre plus accessibles et compréhensibles aux lecteurs.

Au cours des années 1840, l'idée d'intégrer les sténographes dans les services des chambres parlementaires émergea. La Chambre des Pairs fut la première à créer un bureau de rédaction sténographique en 1846, sous la direction d'Hippolyte Prévost. Confrontée à la concurrence, la Chambre des députés suivit le mouvement et intégra les sténographes dans son personnel.

La révolution de 1848 n'interrompit pas cette évolution. Les sténographes furent attachés aux services des assemblées parlementaires, et l'élection des réviseurs par leurs pairs confirma leur intégration. La sténographie était désormais une partie intégrante du service du compte rendu intégral des débats, assurant la publicité des discussions parlementaires conformément au principe constitutionnel.

Le service sténographique de l'Assemblée nationale française doit relever deux défis majeurs liés à la transition de l'oralité à l'écrit dans les débats parlementaires (Gardey, 2010). Tout d'abord, il s'agit de traduire fidèlement et avec précision les discours oraux des orateurs en séance pour produire un texte écrit cohérent. Les sténographes se considèrent comme des experts de l'écrit, chargés de réviser et de clarifier les interventions pour capturer le sens réel des paroles prononcées.

Ensuite, le service sténographique doit produire un document écrit fidèle et authentique, attestant des débats qui ont eu lieu en séance. Le processus de double prise, où les rédacteurs transcrivent les interventions et les réviseurs les corrigent, permet de créer une chaîne de matérialisation du texte, rendant compte des versions successives et des corrections apportées. Cette chaîne de traçabilité est essentielle pour produire une certification des dires et assurer la fiabilité du compte rendu.

De plus, ces traces écrites sont cruciales pour faire face aux éventuelles contestations des élus ou des cabinets. En tant que preuves tangibles, elles permettent de répondre aux réclamations et aux conflits concernant la transcription des discours des députés.

Ainsi, l'organisation hiérarchisée du service sténographique et la production minutieuse du compte rendu sont essentielles pour assurer la sincérité et la validité des débats parlementaires, contribuant ainsi à la confiance du public dans le processus démocratique.

Aujourd'hui encore, bien que la pratique de la sténographie ait été abandonnée au début du XXI^e siècle au profit d'autres méthodes d'enregistrement, le service du compte rendu intégral continue d'assurer la transparence et la publicité des débats parlementaires en France.

1.4. Conserver l'histoire parlementaire

La Division des Archives de l'Assemblée nationale joue un rôle fondamental dans la préservation de la mémoire parlementaire française. En tant que dépositaire des débats, des décisions et des documents officiels, elle assure la pérennité de ce patrimoine unique et offre un accès inestimable à l'histoire politique de la France³. Historiquement, les archives de l'Assemblée nationale étaient initialement confondues avec les Archives nationales. Cependant, avec le temps et l'accroissement du volume de documents, la nécessité d'une division spécifique est devenue évidente. Cette entité distincte a été créée pour gérer, classer et conserver les documents spécifiques à l'Assemblée nationale, reflétant ainsi l'importance accordée à l'histoire parlementaire.

La Division des Archives ne se contente pas de stocker des documents. Elle est chargée de la collecte systématique des documents parlementaires, de leur inventaire, de leur classement et, bien sûr, de leur conservation. De plus, elle joue un rôle crucial de conseil et de formation auprès des services de l'Assemblée, garantissant ainsi que les documents sont correctement archivés dès leur création.

³ État Général Des Fonds Des Archives Nationales (Paris). Série C. 2008,
url:https://www+.siv.archives-nationales.culture.gouv.fr/mm/media/download/Fran_ANX_011185.pdf.

Avec l'avènement de la technologie numérique, la Division des Archives a dû s'adapter pour répondre aux nouveaux défis. Elle a entrepris des initiatives de numérisation pour garantir que les archives soient non seulement préservées, mais également accessibles au public et aux chercheurs du monde entier (Saudrais, 2015). Cette modernisation a également permis de faciliter la recherche et l'accès aux documents, rendant l'histoire parlementaire plus accessible que jamais. L'une des missions essentielles de la Division des Archives est de rendre l'histoire parlementaire accessible au grand public. Elle gère un centre de documentation ouvert à tous, où les citoyens, les chercheurs et les étudiants peuvent consulter des documents parlementaires, des comptes rendus de séances et d'autres archives. La Division des Archives collabore étroitement avec d'autres institutions, telles que la Bibliothèque nationale de France et les Archives nationales, pour partager des ressources et des compétences.

1.5. Numérisation : Modernisation et Accessibilité

À l'ère du numérique, la manière dont nous accédons, conservons et diffusons l'information a radicalement changé. La numérisation des archives parlementaires est devenue une priorité pour l'Assemblée nationale, non seulement pour préserver ces précieux documents, mais aussi pour les rendre plus accessibles au public. Avec le temps, les documents physiques peuvent se dégrader, devenir fragiles ou même disparaître. La numérisation offre une solution pour préserver ces documents de manière durable. De plus, avec l'augmentation constante de la demande d'accès en ligne à l'information, la numérisation est devenue essentielle pour répondre aux besoins du public moderne.

La numérisation des archives parlementaires est un processus méticuleux. Chaque document est soigneusement scanné, catalogué et indexé. Des métadonnées sont ajoutées pour faciliter la recherche et la référence. Les formats numériques utilisés sont choisis pour leur durabilité et leur compatibilité à long terme, garantissant ainsi que les archives restent accessibles malgré l'évolution des technologies. La Division des Archives de l'Assemblée nationale ne travaille pas seule dans cette entreprise. La collaboration étroite avec les institutions telles que la BNF et les archives nationales permettent de partager des ressources, des compétences techniques et des meilleures pratiques pour assurer une numérisation de qualité. Depuis vingt ans, la BNF s'est imposée comme une référence majeure dans la mise en ligne d'images patrimoniales grâce à la numérisation de millions de documents (Sandras, 2020). Actuellement, elle occupe une place centrale dans les pratiques numériques, jonglant entre la mise à disposition d'images pour les utilisateurs en ligne, la préservation du patrimoine, la reconnaissance des droits d'auteur, et les responsabilités des institutions culturelles, incluant l'éducation aux médias et la mise en valeur des documents numérisés.

Grâce à la numérisation, les archives parlementaires sont désormais accessibles à un public bien plus large. Les chercheurs, les étudiants et le grand public peuvent accéder à ces documents

depuis n'importe où dans le monde, à tout moment. De plus, la mise à disposition de ces archives en ligne facilite grandement la recherche, permettant aux utilisateurs de trouver rapidement et efficacement l'information qu'ils recherchent. Les chercheurs peuvent désormais analyser de vastes ensembles de données, explorer des tendances historiques et mener des études comparatives avec d'autres pays.

Lorsqu'on parle de la numérisation massive des textes, on fait souvent référence à la technique de l'OCR, ou "Reconnaissance Optique de Caractères". Cette technologie permet de traiter une image, en l'occurrence un texte numérisé, à travers un moteur OCR. L'intervention de l'Intelligence Artificielle (IA) dans ce processus est déterminante. Elle permet une "traduction" de l'image en texte (Likforman-Sulem, 2003). Concrètement, les pages sont d'abord numérisées, puis transformées en lettres et en mots "discrets", c'est-à-dire distincts et comptables individuellement. Grâce à cette transformation, il devient possible d'explorer ces pages numérisées et de les "différencier" selon différents critères tels que la période, le genre littéraire, la langue, l'année de publication, et bien d'autres.

1.6. Au-delà de la numérisation : Repenser l'interaction et l'éditorialisation des archives numériques

Si la numérisation des archives parlementaires a été une étape décisive pour préserver et rendre accessible le patrimoine historique, il est désormais évident que cette démarche ne peut être une fin en soi. À l'ère du numérique, où l'information est omniprésente et constamment mise à jour, il devient impératif de réexaminer la façon dont nous interagissons avec ces documents numérisés. Dans cette optique, de nouvelles méthodes d'éditorialisation sont envisagées pour maximiser l'utilité et la pertinence de ces archives dans le contexte actuel. Selon Dacos & Mounier (2010)⁴, l'éditorialisation désigne la mise en valeur d'un corpus par diverses actions : sélection de textes, création de collections, établissement d'index thématiques, et mise en avant de focus éditoriaux adaptés aux différents publics. C'est également ce que soutient Bruno BUREAU (2019) qui considère les outils numériques comme un complément essentiel à l'édition papier traditionnelle. Ces outils offrent une variété de possibilités pour élargir la recherche et rendre les travaux philologiques plus accessibles à un public plus large. Pour ce dernier, l'éditorialisation numérique permet une analyse plus approfondie des textes, en particulier lorsqu'il s'agit de traiter des gloses.

Bien que la numérisation offre un accès sans précédent aux archives, elle présente des limites. Les documents numérisés peuvent souvent être perçus comme de simples reproductions

⁴ Dacos, M., & Mounier, P. (2010). III. L'édition au défi du numérique (p. 49-65). La Découverte. <https://www.cairn.info/l-edition-electronique--9782707157294-p-49.htm>

numériques des originaux, sans valeur ajoutée. Avec l'avènement des technologies interactives, il est désormais possible d'engager l'utilisateur de manière plus profonde et significative (Dacos & Mounier, 2010)⁵. Les archives numériques peuvent être enrichies avec des fonctionnalités interactives, telles que des annotations, des liens hypertextes ou des visualisations de données. Ces outils permettent aux utilisateurs d'explorer les documents de manière non linéaire, de suivre des pistes d'intérêt et de découvrir des connexions inattendues.

Les données non structurées, qu'il s'agisse de textes, d'images ou de sons, nécessitent une transformation pour être facilement interprétables et accessibles. C'est là qu'intervient l'encodage, une technique qui transforme ces données en formats spécifiques pour une utilisation optimale.

⁵ Dacos, M., & Mounier, P. (2010). Conclusion / Les cinq piliers de l'édition électronique. À nouveaux enjeux, nouveaux métiers (p. 108-113). La Découverte. <https://www.cairn.info/l-edition-electronique--9782707157294-p-108.htm>

2. Fondements, inspiration et présentation du projet AGODA

2.1. Fondements et inspirations du projet

L'essor d'AGODA est intimement lié à un mouvement global qui cherche à valoriser, exploiter et démocratiser l'accès aux débats parlementaires. Ce mouvement ne se limite pas à la France. En fait, de nombreux projets internationaux ont vu le jour avec des aspirations semblables.

Prenons, par exemple, l'initiative britannique du Hansard⁶. C'est le nom attribué aux retranscriptions des débats dans les gouvernements de type Westminster, que l'on retrouve notamment au Royaume-Uni, mais aussi au Canada et à Singapour. Grâce à une interface web dédiée, le public et les chercheurs peuvent aisément naviguer à travers les comptes rendus des débats du Parlement britannique.

Toutefois, le mouvement ne s'est pas arrêté là. Deux autres projets notables, ParlaClarín et ParlaMint⁷, se sont également démarqués. Leur ambition ? Créer des corpus multilingues des débats parlementaires, le tout encodé en XML-TEI. Ces corpus annotés se veulent être une ressource qui facilite l'échange, la réutilisation et l'analyse profonde des débats parlementaires, quelles que soient les frontières linguistiques. ParlaClarín a mis en œuvre un ensemble d'outils et de méthodologies pour l'annotation et l'analyse de débats parlementaires contemporains. Les méthodologies développées dans le cadre de ce projet ont servi de base solide pour AGODA, en particulier dans la manière dont les débats sont transcrits et structurés. ParlaMint a travaillé sur les transcriptions issues de différentes nations et époques. Les règles de transcription établies dans ce projet ont montré une grande similitude avec celles des débats parlementaires français du XIXe siècle, rendant ainsi sa méthodologie pertinente pour AGODA.

Naomie Truan, lors de sa recherche doctorale, a également emprunté cette voie. Dans sa quête, elle a choisi de mettre l'accent sur l'annotation linguistique en XML-TEI, explorant trois corpus distincts de débats parlementaires issus de la Chambre des communes britannique, du Bundestag allemand et de l'Assemblée nationale française.

Mais pourquoi un tel intérêt pour les débats parlementaires ? La réponse réside dans le potentiel de ces débats à enrichir notre compréhension du discours politique et sociétal. Les projets

⁶ <https://hansard.parliament.uk//>

⁷ <https://aclanthology.org/2020.parlaclarin-1.13/>

mentionnés répondent à un besoin de valoriser cette source d'information auprès des universitaires et du grand public. Les technologies numériques ont joué un rôle prépondérant dans cette dynamique. Elles ont été le pont reliant les débats parlementaires à l'univers plus vaste des humanités numériques, domaine magnifiquement décrit par Pierre Mounier comme une renaissance des sources traditionnelles (Dacos & Mounier, 2010).

Les humanités numériques, dans leur essence, cherchent à établir un dialogue interdisciplinaire autour de l'impact du numérique sur la recherche en sciences humaines et sociales (Bouzidi & Boulesnane, 2017). Ces outils et méthodes numériques révolutionnent la façon dont les chercheurs abordent leurs sources, offrant de nouveaux modes de lecture, élargissant le spectre des sources disponibles et permettant l'émergence de nouvelles questions et perspectives (Longhi, 2017).

Dans ce contexte, les projets comme AGODA ou Hansard, entre autres, ont embrassé le potentiel du numérique. Ils ont vu une opportunité inestimable de renouveler l'intérêt pour les débats parlementaires, offrant de nouvelles méthodes d'analyse et de visualisation qui, espèrent-ils, susciteront une nouvelle vague de recherches et de découvertes.

2.2. Présentation du projet AGODA

Le projet AGODA émerge comme une réponse essentielle aux défis d'accessibilité et d'analyse des débats parlementaires de la Chambre des députés entre 1881 et 1940. Ces débats, conservés dans le Journal officiel de la République française, sont des trésors d'information pour l'histoire politique et sociale, mais sont souvent sous-exploités à cause de leur volumétrie et de leur mode de publication.

L'un des principaux objectifs d'AGODA est d'améliorer considérablement l'accessibilité à ces débats. Pour y parvenir, le projet se consacre à la mise en place d'un workflow qui non seulement rectifie les erreurs potentielles dans les données numérisées, mais ajoute également des annotations pertinentes pour faciliter la recherche.

Le traitement des données joue un rôle crucial dans ce projet. De nombreuses archives historiques sont souvent disponibles uniquement sous forme d'images ou de textes non structurés, ce qui limite grandement leur utilité pour la recherche académique. AGODA s'attaque à ce défi en convertissant ces archives en données textuelles structurées et sémantiquement enrichies. Cela ouvre non seulement ces précieuses informations à un public plus large, mais facilite également leur consultation en ligne.

En rendant ces documents facilement accessibles, le projet entend également faciliter les recherches menées sur ce corpus, permettre la création de sous-ensembles spécifiques de données pour des études ciblées et introduire des modalités innovantes de visualisation de ces

documents. Au-del  d'une lecture d taill e, dite "proche", le projet pr voit d'introduire des m thodes de "lecture distante", comme la mod lisation de sujets ou l'analyse de r seaux.

Techniquement, plusieurs outils et m thodes ont  t  utilis s pour r aliser les objectifs du projet. La reconnaissance optique de caract res (OCR) transforme les images scann es en textes exploitables, et le traitement automatis  des langues apporte une segmentation, une annotation et un enrichissement s mantique aux donn es. Le choix de coder et structurer les textes en utilisant la norme XML-TEI garantit que, tout en  tant ais ment consultables par des machines, les donn es conservent leur riche contexte s mantique (Verlaet, 2010). Le processus d'annotation est au c ur de cette d marche. Il ne se contente pas uniquement de mettre en lumi re la structure des comptes rendus, mais aussi de souligner et d'extraire les informations pertinentes des d bats pour des comparaisons ult rieures. Cette annotation, con ue dans une optique de donn es li es, vise  galement   associer, lorsque possible, les entit s nomm es   des identifiants uniques (URI) fournis par la Biblioth que nationale de France. En annotant ces d bats, AGODA cherche   rendre ces informations plus exploitables et r utilisables.

Finalement, l'intention est de publier ces donn es sur une plateforme intuitive en utilisant une base de donn es eXist et l'application TEI Publisher.

2.3. Objectifs principaux

1. **Accessibilit  et Exploitation** : Le projet vise   offrir un acc s enrichi   ces donn es. Cela signifie non seulement corriger et enrichir les donn es num ris es, mais aussi rendre ces donn es interop rables et fournir des outils pour les exploiter (comme la fouille de textes, l'analyse et l'indexation).
2. **Structuration et Enrichissement S mantique** : L'objectif est de transformer les images et textes bruts actuellement disponibles en donn es textuelles structur es et s mantiquement enrichies. Ceci est essentiel pour permettre une exploration en ligne plus efficace et des analyses plus approfondies.
3. **"Preuve de concept"** : Plut t que de traiter l'ensemble du corpus en une fois, AGODA commencera par travailler sur une sous-section, la Ve l gislation de la Troisi me R publique (1889-1893). Cela servira de prototype et permettra de concevoir un flux de travail qui pourrait  tre appliqu    d'autres projets similaires.

2.4. La ch ne de traitement du projet (Workflow)

L' ditorialisation et l'enrichissement des d bats parlementaires n cessitent la conception d'un workflow adapt    la production et   l'analyse de tels grands corpus de documents historiques. Selon Lipsyc & Ihadjadene (2013), la dimension op ratoire de l' ditorialisation englobe une

"chaîne de production", c'est-à-dire une série d'opérations indispensables pour concrétiser un "produit" documentaire, souvent désignée comme "chaîne éditoriale".

AGODA s'appuie sur une série d'étapes successives, un processus défini pour traiter des documents afin d'atteindre les objectifs prédéfinis. Ce processus est conçu pour traiter de vastes ensembles de documents historiques, s'inspirant du modèle mis en place pour le projet ANR TIME-US (2018-2021). Ce dernier avait pour mission de gérer un large ensemble de documents historiques variés, afin d'améliorer l'accessibilité, la recherche, et la visualisation. AGODA, ayant des ambitions similaires, a adopté et adapté ce modèle à ses propres besoins.

Le processus d'AGODA se décompose en quatre étapes clés :

- **Reconnaissance optique de caractères** : L'extraction des textes à partir d'images est une tâche complexe qui requiert une stratégie précise. Pour maximiser l'efficacité, il est envisagé de combiner deux approches : ré-océrer et post-corriger les textes avec le plus d'erreurs, et simplement corriger les textes déjà océrés mais moins fautifs. L'ambition est de progresser dans l'annotation en TEI sans attendre que l'ensemble du corpus soit parfaitement corrigé.

Cependant, plusieurs défis ont été rencontrés, notamment la qualité de la numérisation des documents et les défis liés à l'océration elle-même. Bien que la Bibliothèque nationale de France ait fourni des textes avec un taux de reconnaissance élevé, l'équipe a découvert des problèmes de qualité sur certaines pages. Ces problèmes étaient dus à des facteurs tels que la résolution, le contraste, la luminosité, l'inclinaison des pages et la qualité du support physique des documents.

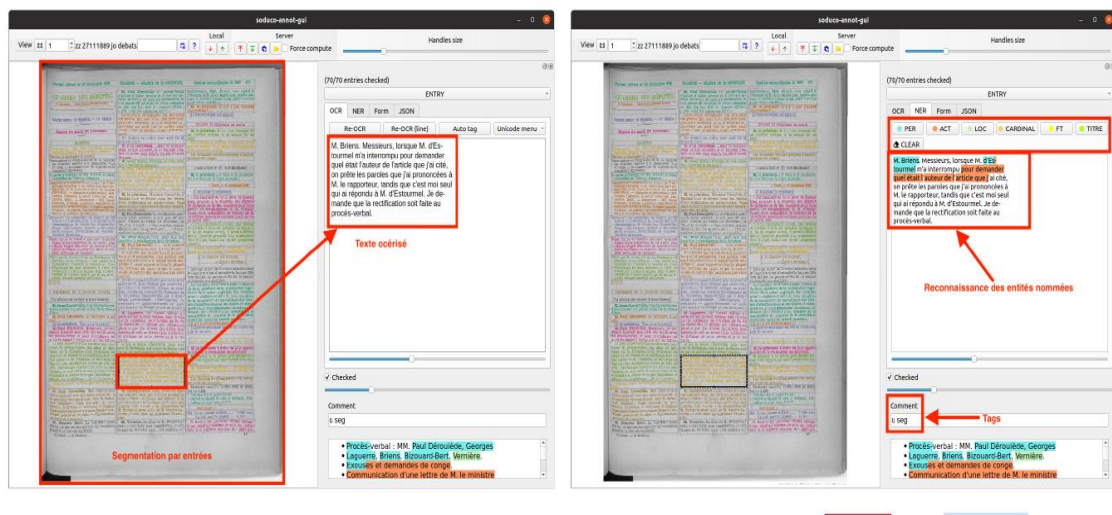


Fig. 1 : Outil LRDE ; phase d'océration et d'annotation en commentaire

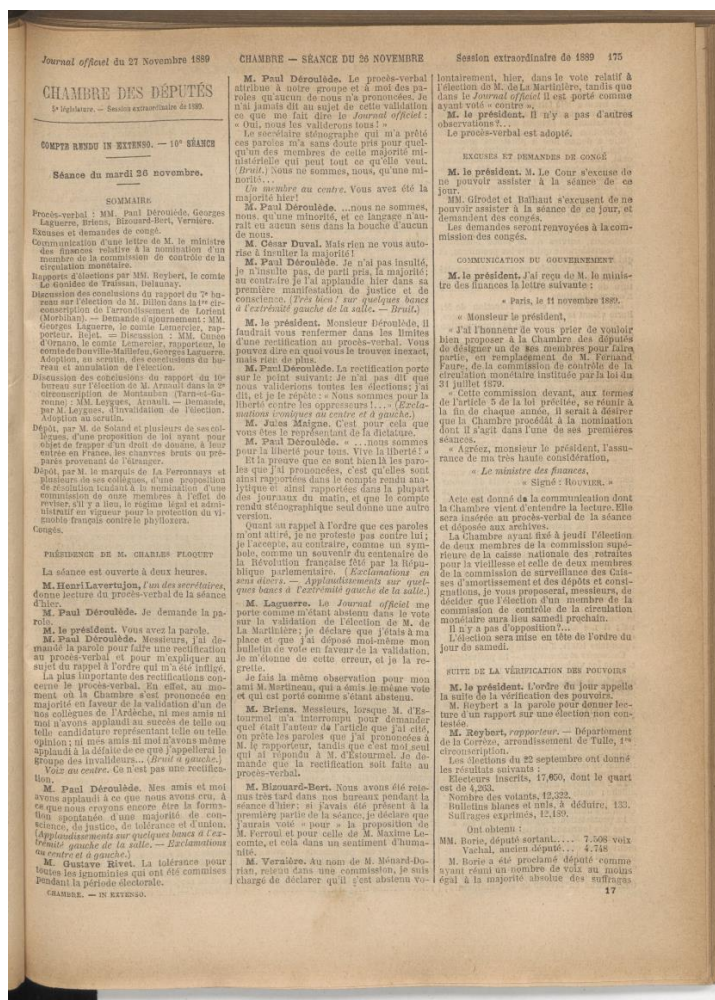


Fig. 2. Première page du compte rendu in extenso de la séance du 26 novembre 1889 (Gallica)⁹

Pour surmonter ces difficultés, l'équipe a essayé d'améliorer la qualité des images, notamment par le redressement des images. Ils ont également exploré différentes méthodes et outils pour la ré-océrisation. Finalement, ils ont opté pour l'outil d'OCR du LRDE⁸, développé spécifiquement pour les textes historiques. Cet outil a offert de nombreux avantages, dont une identification précise des entités nommées et une interface utilisateur intuitive.

Par ailleurs, l'équipe AGODA réfléchit à une solution innovante comme le développement d'une Intelligence Artificielle, spécifiquement une SVM, capable de trier le corpus en "textes peu fautifs" et "textes très fautifs". L'évaluation du taux d'erreur entre différentes plateformes, comme Gallica, OCR Tesseract, Abby et eScriptorium, est primordiale. La création d'un ground truth et l'entraînement d'un modèle avec eScriptorium sont en cours.

Post-correction : Comme aucune solution OCR n'est parfaite, il est indispensable de corriger les erreurs qu'elle pourrait introduire. Une phase de post-correction est donc mise en œuvre en utilisant la bibliothèque Python pypellchecker. Cependant, en raison de la spécificité du corpus, une adaptation de cette étape est

⁸ Laboratoire de Recherche et Développement (LRDE)

⁹ url vers les fichiers numérisés : <https://gallica.bnf.fr/ark:/12148/bpt6k64948012>

nécessaire, notamment en utilisant un dictionnaire basé sur des textes français du XIX^e siècle.

- **Annotation en XML-TEI** : L'étape suivante concerne l'annotation et l'amélioration sémantique des données. C'est une méthode standard pour coder des textes de manière à les rendre lisibles par une machine tout en conservant les informations sémantiques et structurelles pertinentes. Cette étape permet de structurer les informations, de rendre les textes plus interopérables et d'enrichir les données. Un schéma XML adapté est élaboré pour répondre aux besoins spécifiques du projet. Cela commence par la définition des méthodes d'encodage et le choix des outils techniques appropriés. L'application de ces méthodes sur les textes nécessite une automatisation, étant donné la taille massive du corpus.

- **Publication en ligne avec eXist-db et TEI Publisher** : Les textes annotés sont ensuite stockés dans une base de données eXist-db. Le TEI Publisher, à son tour, est utilisé pour convertir ces données en pages web HTML, rendant ainsi les débats parlementaires disponibles en ligne. Cette interface permet une navigation aisée, une recherche en texte intégral et l'affichage de fac-similés.

- **Modélisation de sujets et intégration sémantique** : Pour ajouter de la profondeur et de la pertinence à l'ensemble du corpus, des techniques telles que la modélisation de sujets (ou "topic modeling") sont utilisées. Les résultats sont ensuite intégrés au corpus sous forme d'annotations sémantiques.

Analyse des données : La dernière phase est dédiée à l'examen des données traitées. Elle peut s'intégrer au processus global ou se dérouler après la mise en ligne des données. Cela concerne l'analyse initiale pour améliorer la présentation des données et les futures analyses menées par les utilisateurs grâce à une API.

Importance pour la recherche :

Ce projet intéresse non seulement les historiens mais aussi d'autres domaines comme le droit, les sciences politiques, la sociologie et la linguistique. Les débats parlementaires offrent un aperçu inestimable de l'évolution politique, sociale et culturelle de la France pendant cette période.

2.5. Porteurs du projet

Le projet AGODA est le reflet d'une collaboration riche et multidisciplinaire, s'appuyant sur des compétences variées et complémentaires pour mener à bien ses objectifs ambitieux. Avec un partenariat majeur de la BnF via le DataLab, ce projet est soutenu tant sur le plan financier que technique, ce qui lui donne une solide assise pour son développement.

- Le **DataLab de la BnF** joue un rôle essentiel dans la mise en œuvre du projet AGODA. Grâce à son soutien financier, technique et scientifique, il permet au projet d'avoir accès à des ressources essentielles et à un accompagnement spécialisé pour garantir son bon déroulement.

Les équipes participantes sont la preuve de la diversité des compétences mobilisées :

- **LARHRA** apporte une expertise en histoire, et plus particulièrement en histoire numérique. Pierre VERNUS, membre de ce laboratoire, joue un rôle pivot dans la coordination et la direction du projet.

- **MNSHS** injecte une dimension numérique au projet, en combinant des compétences en humanités numériques et en informatique, avec des acteurs clés tels que Marie Puren, Nicolas Bourgois et Aurélien Pellet.

- **L'Inria-ALMAAnaCH**, avec son expertise en traitement automatique des langues, est essentiel pour la compréhension et l'analyse des textes du corpus. Julien Martin et Éric de la Clergerie représentent des piliers technologiques du projet.

- L'implication **des équipes en appui** renforce le projet en offrant une expertise complémentaire. Le **LRDE de l'EPITA**, avec son expérience en traitement d'images, contribue directement à la phase d'océrisation. De son côté, la Division des Archives et de l'Histoire parlementaire de l'Assemblée nationale offre une profondeur historique et technique, crucial pour assurer l'exactitude et la pertinence des données traitées.

Le projet AGODA est une initiative collaborative qui mobilise un large éventail de compétences pour traiter et valoriser un corpus de documents historiques de grande envergure. Sa réussite repose sur cette synergie entre acteurs divers, tous dédiés à la réalisation des objectifs fixés.

En résumé, AGODA est une initiative passionnante qui promet de transformer la manière dont les chercheurs et le grand public accèdent et explorent un trésor d'informations sur la politique et la société françaises de la fin du XIXe et du début du XXe siècle. AGODA aspire à innover dans la manière dont les grands corpus de documents historiques sont analysés. En élaborant un processus de travail adapté à l'analyse de tels volumes de données, le projet espère offrir une méthodologie qui peut être adoptée et réutilisée dans d'autres initiatives. Cette démarche vise également à produire des données conformes aux principes FAIR, en utilisant les langages standards de l'informatique, en documentant méticuleusement les étapes et procédures et en garantissant l'accès ouvert au code source des outils développés.



Fig. 3 – Illustration de la chaîne de traitement

2.6. Les projets similaires et les avancées dans le domaine

Dans le domaine en pleine expansion des humanités numériques, de nombreux projets ambitieux cherchent à révolutionner la manière dont nous appréhendons, conservons et diffusons le patrimoine culturel et historique.

Les projets qui vont être présentés dans cette section ne représentent qu'un échantillon non exhaustif des initiatives en cours dans le domaine des humanités numériques ; ils sont issus exclusivement des interventions partagées lors de la journée d'études¹⁰ « retour d'expérience en édition numérique des textes ». L'objet de cette journée d'Étude est de mettre en relation des historiennes et des historiens travaillant sur l'édition numérique pour confronter leurs projets, leurs questionnements, leurs réussites et leurs échecs. Cette journée d'étude est organisée par Les Éditions Chrétiens & Sociétés, conjointement avec l'Axe de Recherche en Histoire Numérique du LARHRA qui prône l'encodage sémantique des données. Ces deux derniers s'inscrivent dans le mouvement numérique en matière d'édition. Adoptant les principes de la Science Ouverte, ils se tournent vers l'édition numérique des sources, traitant ainsi les documents imprimés comme s'ils étaient des bases de données.

Les projets suivants, présentés lors de la journée de retour d'expérience, illustrent cette tendance :

Editer un texte à l'ère du numérique par Ariane Pinche (CNRS)

La transition vers le numérique a bouleversé les paradigmes traditionnels de l'édition. Ariane Pinche, du CNRS, éclaire ce phénomène en évoquant les particularités des éditions numériques. À la différence des éditions traditionnelles, elles permettent de stocker et de manipuler plusieurs strates d'informations. Les éditions axées sur XML TEI émergent comme des outils particulièrement efficaces pour interroger et réutiliser les données. Ariane Pinche en est un exemple vivant, ayant mis en œuvre une édition nativement numérique pour le recueil hagiographique "Li Seint Confessor" de Wauchier de Denain, démontrant ainsi la valeur ajoutée de cette approche. Cette démarche lui a valu de nombreux éloges et récompenses, soulignant l'importance de la numérisation dans les recherches actuelles en sciences humaines.

Corpus calibrés pour l'histoire de la langue française par Pierre Larrivée et Natasha Romanova (Université de Caen)

Le français, riche et évolutif, est le sujet d'étude de Pierre Larrivée et Natasha Romanova. Ils ont introduit trois corpus historiques : ConDÉ, Micle, et Chroniques, qui servent de fenêtres sur l'évolution de la syntaxe française à travers les âges. Chaque corpus est précieux, mais ils notent

¹⁰ Journée d'études du 22 juin 2023, Lyon

que l'annotation manuelle est capitale pour la précision, même si ConDÉ est déjà annoté en POS. Les travaux respectifs de Larrivée, axés sur l'interprétation des configurations grammaticales, et de Romanova, centrés sur l'édition numérique et l'annotation des corpus, suggèrent que la langue française, tout en étant ancrée dans une riche tradition, est également façonnée par les innovations technologiques.

Éditer la recherche et les données de la recherche par Christine Chadier (Université Lyon 3)¹¹

Face au besoin croissant de publier à la fois les données de recherche et leurs résultats, des solutions adaptées se dessinent. Les critères FAIR (Faciles d'accès, Accessibles, Interopérables, Réutilisables) guident cette évolution. Pour les publications associées aux "Chrétiens et Sociétés, Documents" et à la revue "Chrétiens et Sociétés XVIe-XXIe siècles", la plateforme Nakala, gérée par Huma-Num en collaboration avec l'IN2P3/CNRS, a été choisie. Cette solution favorise le stockage sécurisé, l'usage de formats ouverts et facilite la référencement des données dans les publications d'OpenEdition Journals ou OpenEdition Books. En poursuivant l'aspiration d'une science ouverte, une fusion entre plateformes de publication et entrepôts de données a été initiée. Au-delà d'une simple "politique de données", l'objectif est de centraliser et d'uniformiser les dépôts, envisageant à terme une plateforme de valorisation avec Nakala_Press.

Christine Chadier, experte en édition de sources en XML-TEI, joue un rôle central dans cette démarche, combinant expertise historique et compétences techniques

Faciliter l'édition numérique avec les méthodes de reconnaissance automatique de textes par Angela Göbel (Université Lyon 3, LARHRA Herzog August Bibliothek Wolfenbüttel)

À l'intersection de l'édition numérique et de la reconnaissance automatique de textes, Angela Göbel met en avant l'outil Transkribus dans le cadre du projet "Grand Tour Digital". Cette initiative, au cœur de la recherche contemporaine, vise à numériser des témoignages historiques, notamment ceux de voyages. L'utilisation de Transkribus révolutionne la manière dont les textes sont traités, permettant une transcription semi-automatique des manuscrits. Cette fusion de technologie et de recherche historique ouvre des portes à une compréhension plus approfondie et diversifiée des récits de voyage, comme celui de Wagener.

À la découverte des humanités numériques par Olivier Spina (Université Lyon 2, LARHRA)

Olivier Spina nous invite à un voyage dans le temps, à la redécouverte d'une enquête réalisée en 1543. Cette enquête, conservée précieusement à la Parker Library, jette un éclairage unique

¹¹ C. Chadier, L'édition des Actes des synodes des églises réformées de Bourgogne au XVIIe siècle.

<https://shs.hal.science/halshs-01652043/document>

sur les croyances et les pratiques religieuses de l'époque (sous le règne d'Henri VIII). Dans son travail, Spina met en avant les défis et les opportunités présentés par les humanités numériques. La numérisation de tels documents n'est pas seulement un exercice technique, mais elle implique aussi une réflexion approfondie sur la manière de rendre ces informations pertinentes et accessibles pour le public moderne.

De la transcription de travail à la publication numérique par Francesco Beretta et Morgane Pica

Francesco Beretta et Morgane Pica nous immergent dans l'univers intime d'Anna Maria Preiswerk-Iselin, à travers ses journaux personnels. Ces écrits, témoignant de la vie d'une femme du 19^e siècle, nécessitent une transcription rigoureuse. Grâce à l'outil Geovistory, cette transcription a été convertie en une base de données XML/TEI, enrichie par des annotations sémantiques. Le duo détaille leur méthodologie, soulignant l'importance de l'encodage pour la recherche en sciences humaines et la pertinence d'une base de données interconnectée pour des études plus approfondies.

Point commun entre ces projets

L'examen de ces projets révèle un point commun essentiel : l'utilisation innovante des technologies numériques pour préserver, analyser et partager le patrimoine culturel et historique. Qu'il s'agisse de convertir des textes imprimés en bases de données interactives, de numériser des enquêtes anciennes ou de transcrire des journaux intimes, tous s'efforcent d'accroître l'accessibilité et la compréhension des trésors historiques. Ces initiatives montrent l'évolution du paysage de la recherche en humanités numériques, mettant l'accent sur l'accessibilité, l'interopérabilité, et la réutilisabilité des données. Le passage vers l'édition numérique, accompagné par l'adoption de standards comme XML-TEI et la prise en compte des critères FAIR, illustre la volonté d'aller vers une science plus ouverte et collaborative.

Problématique

Le projet AGODA, visant à exploiter au maximum les capacités des humanités numériques pour dépasser les frontières traditionnelles de la recherche historique et culturelle, se trouve à la croisée des avancées technologiques et méthodologiques.

Au cœur de ses aspirations se trouve l'ambition de traiter une quantité massive de documents textuels, et pour ce faire, l'automatisation du balisage XML-TEI se révèle cruciale. Optimiser ce processus permettrait non seulement d'accélérer considérablement le traitement des documents, mais également de garantir une cohérence et une standardisation, essentielles pour les analyses ultérieures et la mutualisation des données. C'est donc à cet effet que j'ai souhaité libeller la problématique comme suit :

« Dans quelle mesure est-il possible d'automatiser entièrement le processus de balisage XML ? Comment optimiser l'automatisation du balisage XML-TEI ? »

Cette problématique reflète directement les enjeux auxquels AGODA est confronté. De plus, dans un contexte où la qualité et l'intégrité des données sont primordiales, l'automatisation, lorsqu'elle est bien exécutée, offre une fiabilité accrue, réduisant les erreurs humaines potentielles ainsi qu'un gain de temps. Pour AGODA, maîtriser cette étape d'automatisation signifie donc se doter d'un outil puissant pour atteindre ses objectifs ambitieux d'analyse et de diffusion du patrimoine historique et culturel à une échelle sans précédent.

3. Mise en œuvre de l'encodage : les enjeux du balisage automatique.

3.1 La conversion des fichiers : De la notation JSON à l'encodage XML-TEI

L'automatisation de l'encodage a été conçue en tenant compte du format initial des données. Le format d'une donnée définit comment cette dernière est représentée ; c'est une norme structurelle qui permet une interprétation par divers programmes et logiciels. En utilisant l'outil OCR du LRDE sur les images numérisées des débats parlementaires, nous avons obtenu des données textuelles au format JSON. L'outil OCR avait été préconfiguré pour produire des résultats dans ce format. Le JSON structure ses données sous forme de paires clé-valeur à l'intérieur d'une liste ordonnée. C'est un format interopérable, facile à partager et à convertir.

Chaque fichier JSON est structuré comme une liste d'éléments, chaque élément étant un dictionnaire avec plusieurs clés.

Attributs Clés :

box : Chaque élément possède un attribut "box" qui représente des coordonnées ou des dimensions liées à la position d'un élément sur une page.

id : Un identifiant unique attribué à chaque élément.

parent : Cela indique l'élément parent ou le niveau hiérarchique précédent de l'élément actuel, créant ainsi une structure arborescente ou hiérarchique.

text_ocr : C'est le texte extrait ou reconnu associé à l'élément. Il peut s'agir du texte d'un titre, d'un paragraphe ou d'une autre section du document.

ner_xml : C'est le même texte que « text_ocr » qui est extrait ou reconnu associé à l'élément. La seule différence est que celui-ci est enrichi grâce à la reconnaissance d'entité nommées avec des balises tels que <PER> et <LOC>.

type : Indique le type de l'élément, comme "PAGE", "TITLE_LEVEL_1", "SECTION_LEVEL_1", etc. Cela donne une idée de la structure ou de la hiérarchie du document.

comment (s'il existe) : Fournit des commentaires ou des annotations supplémentaires pour l'élément.

L'attribut "comment" est la clé spécifique, le pilier central sur lequel nous bâtissons le script. Elle joue un rôle prédominant car c'est à partir de sa valeur que nous définissons comment une balise spécifique doit être structurée. En d'autres termes, chaque "comment" nous guide sur la manière dont une information est censée être interprétée et encodée. Les fichiers JSON contiennent les résultats obtenus grâce à l'OCR et sont structurés tels qu'on peut le voir sur l'image en dessous. Ces fichiers renferment trois types d'informations majeures, reflétant trois phases clés du processus d'océrisation effectué par l'outil du LRDE : la segmentation, la reconnaissance des caractères, l'enrichissement des données (reconnaissance des entités nommées, annotation).



Le code JSON ci-dessous est illustré avec des annotations en français :

- Début de l'objet** : Pointe vers l'ouverture de la accolade {.
- Liste**- Couple clé / valeur** : Pointe vers le champ "comment" et sa valeur "u-beginning seg".
- Fin de l'objet** : Pointe vers la fermeture de la accolade }.

```
{
  "activities": [],
  "addresses": [],
  "box": [
    60.572994652541155,
    1786.0,
    557.4270053474592,
    106.86131907344293
  ],
  "checked": true,
  "comment": "u-beginning seg",
  "id": 307,
  "key": [
    0,
    1839
  ],
  "ner_xml": "<PER>M. Paul Déroulède</PER>. Messieurs, <ACT>j</ACT>'<ACT>ai de-<0x2029>mandé la parole pour faire une rectification-<0x2029>au pro",
  "origin": "computer",
  "parent": 269,
  "persons": [
    "M. Paul Déroulède"
  ],
  "text ocr": "M. Paul Déroulède. Messieurs, j'ai de-\nmandé la parole pour faire une rectification\nau procès-verbal et pour m'expliquer au\nsuj",
  "type": "ENTRY"
}
```

Fig. 4 : Extrait d'un fichier JSON ; résultat de l'océrisation

Un fichier JSON correspond à une page numérisée du journal. Une séance est décrite sur plusieurs de ces pages (fichier JSON). Par conséquent, l'ensemble des fichiers JSON d'une séance est regroupé dans un unique dossier. La tâche consiste à fusionner tous ces fichiers JSON pour reconstituer la séance sous forme d'un seul fichier XML. Une fois cette conversion réalisée, chaque fichier XML est validé selon le schéma Relax RNG. Ces fichiers XML validés sont ensuite stockés dans un dossier dédié. En phase finale, tous ces fichiers XML sont amalgamés en un unique fichier, nommé "Fichier XML Corpus".

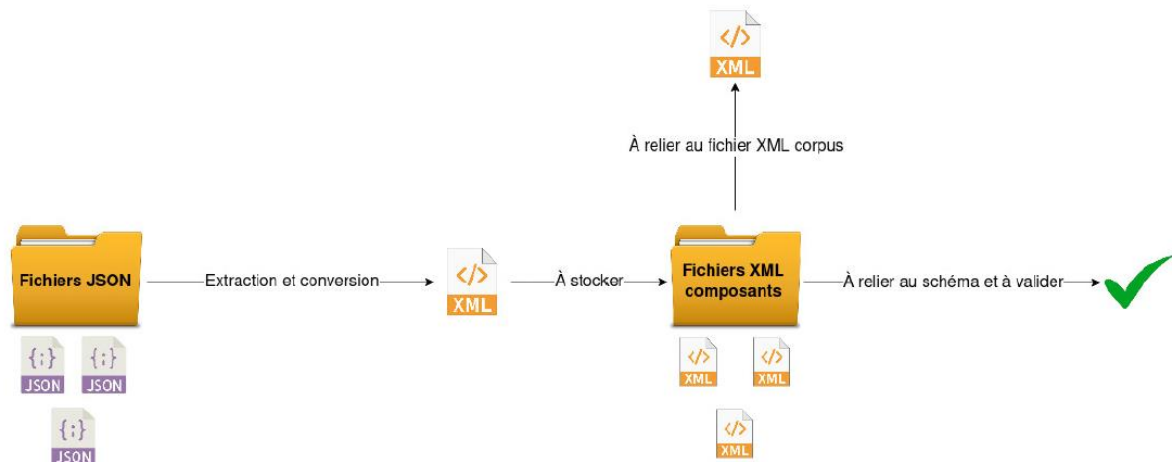


Fig. 5 : chaîne du traitement automatique

3.2 L'adoption du XML-TEI pour le projet AGODA

XML, ou eXtensible Markup Language, est un langage de balisage qui définit un ensemble de règles pour encoder des documents de manière à ce qu'ils soient à la fois lisibles par l'homme et par la machine. Il est un outil crucial pour l'éditorialisation numérique des textes anciens, en particulier dans le domaine philologique (Bureau, 2019). Voici quelques raisons pour lesquelles XML est largement utilisé :

Interchangeabilité : XML est souvent utilisé pour échanger des données entre différents systèmes, car il est à la fois standardisé et extensible.

Indépendance de la plateforme : Étant un format texte, XML est indépendant de toute plateforme ou langage de programmation.

Flexibilité : Contrairement à d'autres formats de données, XML permet aux utilisateurs de définir leurs propres balises, offrant une grande flexibilité pour représenter des structures de données complexes.

Support étendu : De nombreux outils, bibliothèques et systèmes de bases de données offrent un support natif pour XML.

Le XML utilise des balises encapsulées dans des chevrons, permettant de définir le rôle intellectuel des mots ou groupes de mots au sein d'un texte. Ces balises, qui délimitent différents éléments XML, peuvent être affinées avec des attributs. L'une des particularités majeures du XML est sa capacité à organiser les éléments de manière hiérarchique, sous forme d'arbre, permettant à certains éléments d'hériter des propriétés de leurs parents.

Cependant, le XML en lui-même ne spécifie pas le nom des balises. C'est ici que la Text Encoding Initiative (TEI) entre en jeu. Fondée par des chercheurs du Vassar College en 1987, la TEI est basée sur le langage XML et propose un ensemble de balises prédéfinies pour décrire les textes de manière scientifique et sémantique. Elle a évolué au fil du temps pour devenir un standard incontournable dans les humanités numériques, avec une cinquième version appelée TEI P5.

La TEI a été conçue pour être suffisamment flexible pour s'adapter à divers types de textes, quelle que soit leur forme, date ou langue. Elle est organisée en modules, qui rassemblent des balises communes, et en classes, qui organisent les éléments selon leur fonction ou position (Burnard, 2015). Cela offre aux chercheurs et éditeurs une grande diversité dans la manière d'encoder leurs textes.

Lors de sa conception, le projet AGODA a décidé d'utiliser le format XML-TEI pour encoder son corpus, ce choix n'étant pas anodin. En effet, le XML-TEI est plus qu'un simple langage de balisage. Né du World Wide Web Consortium en 1998, il est issu du SGML et a pour objectif de structurer les données de manière à être compréhensibles tant pour l'homme que pour la machine (Verlaet, 2010). De ce fait, il facilite grandement l'échange de données entre différentes plates-formes.

Le choix du XML-TEI s'est imposé pour plusieurs raisons. La première étant que la TEI est conçue spécifiquement pour les chercheurs en sciences humaines et sociales. Elle offre un large éventail de balises qui répondent aux besoins diversifiés des chercheurs, tout en étant soutenue par une communauté dynamique. De plus, elle est régulièrement mise à jour, garantissant ainsi sa pertinence et sa longévité.

En outre, le XML-TEI facilite l'échange de données, un aspect essentiel pour AGODA. Sa nature ouverte et son indépendance logicielle le rendent interopérable, favorisant la collaboration. De plus, la TEI permet différents niveaux d'encodage, qu'ils soient visuels, sémantiques ou analytiques, offrant une grande flexibilité dans la manière dont les données sont présentées et analysées.

Enfin, tout encodage doit respecter certaines règles pour être considéré comme valide. Dans le cas du XML-TEI, cela implique la conformité à la fois à la syntaxe XML et aux spécifications de la TEI. Cela signifie que chaque document doit commencer par une déclaration XML et un élément racine TEI, et doit également contenir au moins un en-tête et un élément texte.

La mise en œuvre de l'automatisation nécessite une compréhension approfondie de la structure des fichiers JSON. Ces fichiers dictent la manière dont l'information est organisée. Par ailleurs, pour assurer la cohérence et la précision de cette automatisation, il faut s'appuyer sur le guide

d'annotation¹² élaboré méticuleusement par Fanny en collaboration avec Madame Puren. Ce guide sert de boussole pour naviguer à travers les méandres des données et garantir un marquage adéquat.

3.3. L'Adoption de l'ODD

L'ODD, abréviation pour "One Document Does it all", est un concept formulé par Lou Burnard et Michael Sperberg-McQueen lors de la phase initiale du développement de la TEI. Ce fichier XML-TEI sert à documenter et formaliser les options d'encodage. Littéralement traduit, cela signifie "un seul document fait tout" ou "document tout-en-un". Il procure une information technique essentielle pour divers processus informatiques, tout en offrant une documentation pour consultation humaine.

Avec un ODD, on peut produire de multiples résultats : des schémas de validation en plusieurs langages, une documentation exhaustive sur tous les composants utilisés dans le schéma, et une documentation descriptive mettant en lumière les décisions d'encodage.

L'ODD est structuré comme un fichier XML-TEI, obéissant à la syntaxe XML et aux balises TEI. Il se caractérise par une organisation hiérarchique, comprenant des éléments, des attributs et des valeurs. Les balises proviennent spécifiquement du module "tagdocs". Le document se compose essentiellement de trois parties : les métadonnées, le contenu principal de la documentation, et les spécifications d'encodage.

L'ODD¹³ a été rédigé par Fanny LEBRETON pendant son stage sous la supervision de Marie PUREN et Pierre VERNUS.

L'ODD s'est imposé comme l'outil idéal pour répondre aux besoins multiples. L'équipe AGODA aspire à produire un encodage XML-TEI qui soit non seulement conforme mais également valide, en harmonie avec la syntaxe XML et les directives clés de la TEI. De plus, il était important de documenter et formaliser nos choix d'encodage via un schéma adapté, et c'est exactement la solution que propose l'ODD dans sa richesse fonctionnelle.

Un encodage est considéré valide lorsqu'il s'aligne sur un schéma prédéfini qui en établit la structure, et plusieurs formats peuvent servir à ce dessein. Grâce à la TEI, il est possible de personnaliser cet encodage pour le rendre conforme à des besoins spécifiques, garantissant ainsi une utilisation cohérente et une potentielle réutilisation par d'autres acteurs du projet.

¹² https://github.com/FannyLbr/Memoire-AGODA-TNAH2022/blob/main/C%20-%20Encodage%20automatique/C1%20-%20Guides/guide_annotations_agoda.pdf

¹³ https://agoda-project.github.io/agoda_odd.html

Intrinsèquement, l'ODD est profondément ancré dans l'écosystème de la TEI. Construit en XML-TEI, sa pertinence est telle qu'il se voit dédier un chapitre entier dans les lignes directrices de la TEI¹⁴. Ainsi, en adoptant l'ODD, nous assurons une standardisation optimale de la communication de nos choix d'encodage.

La pérennité de notre travail est également renforcée par l'utilisation d'un format standard comme l'ODD, d'autant plus qu'il bénéficie de mises à jour régulières.

Et enfin, l'ODD présente l'avantage considérable d'être polyvalent : il peut être transformé en de nombreux formats, rendant le document source plus accessible et adaptable aux différentes exigences des projets.

3.4. Rappel sur l'objectif du stage et travail réalisé

L'automatisation s'est imposée comme une méthode efficace pour encoder l'ensemble du corpus sans avoir à manuellement insérer les balises. Au cours de ma période de stage de quatre (04) mois, l'unique tâche importante qui m'a été confié consistait donc à élaborer un programme, c'est-à-dire un ensemble d'instructions structurées, pour gérer cet encodage. L'objectif était de générer un fichier XML qui se rapproche le plus possible du modèle¹⁵ "ideal encoding". Ce fichier de référence est unique car il représente le standard d'encodage souhaité, ayant été balisé manuellement avec précision et soin. Je devais ensuite ajuster le programme en fonction des résultats fournis par la machine. Après avoir réussi cette étape, il est prévu de soumettre le programme à des tests en utilisant d'autres ensembles de données regroupés par séance. L'objectif était de s'assurer que le script était adaptable et efficace dans différents contextes et avec différentes données.

L'approche initiale adoptée pour cette tâche a été réalisée par la stagiaire précédente. Son script était principalement basé sur l'utilisation d'expressions régulières, complétées par un mécanisme de remplacement¹⁶. Cependant, malgré ses efforts, les résultats obtenus n'étaient pas à la hauteur des attentes. En effet, la phase de validation selon le schéma Rng présentait des difficultés pour certains fichiers. Certains fichiers XML générés ne correspondaient pas suffisamment au fichier "ideal encoding" et n'étaient pas valide selon le schéma rng, soulignant la nécessité d'une nouvelle approche.

¹⁴ TEI Guidelines

¹⁵ https://github.com/FannyLbr/Memoire-AGODA-TNAH2022/blob/main/B%20-%20Mod%C3%A9lisation%20XML-TEI/B2%20-%20Encodage%20test/FR_3R_5L_1889-11-26_ideal_encoding_model.XML

¹⁶ Voir <https://github.com/FannyLbr/Memoire-AGODA-TNAH2022/tree/main/C%20-%20Encodage%20automatique/C2%20-%20Scripts%20Python>

C'est dans ce contexte que ma mission a débuté. Plutôt que de m'appuyer principalement sur des expressions régulières, j'ai décidé d'adopter la bibliothèque LXML qui m'a été suggéré pour cette tâche. Cette librairie python, reconnue pour sa robustesse et sa flexibilité, m'a offert les outils nécessaires pour traiter, analyser et modifier des documents XML de manière plus structurée et précise.

Mon objectif principal était d'améliorer considérablement la qualité de l'encodage automatique et de produire un fichier XML qui se rapproche le plus possible du standard établi par le fichier "ideal encoding" et valide selon le schéma Rng. A la fin du stage, bien que les résultats obtenus aient été conformes au schéma RNG, les fichiers XML produits n'étaient pas tout à fait à la hauteur de l'idéal envisagé. En effet, ils manquaient certaines informations telles que l'URI. Cette absence est notable car une bonne annotation devrait s'aligner sur une approche de données liées (linked data), en identifiant autant que possible les entités nommées (comme les orateurs, les lieux et les organisations) en utilisant les URI fournis par la BnF (informations dont nous ne disposons pas pendant cette période).

Par ailleurs, les fichiers XML générés comportaient des lacunes et certaines ambiguïtés. Dans les sections à venir, je décrirai et évaluerai la méthode d'automatisation que j'ai développée, en tenant compte des spécificités de la source et des défis techniques.

3.5. Choix du langage et des librairies pour le traitement automatique

Parmi les nombreux outils disponibles pour l'encodage automatique, LXML se distingue comme une bibliothèque puissante pour le traitement XML et HTML. Offrant une interface facile à utiliser, basée sur les bibliothèques C libXML2 et libxslt, elle permet une manipulation rapide et efficace des documents. Outre LXML, il existe d'autres technologies et méthodologies qui ont été adoptées au fil des ans, chacune avec ses propres avantages et inconvénients.

Ses avantages principaux comprennent :

Performance : Traitement rapide grâce à sa base en langage C.

Compatibilité : Interface compatible avec la bibliothèque standard ElementTree de Python.

Fonctionnalités étendues : Support pour XPath, XSLT, validation avec schémas XML, et bien plus.

Robustesse : Capable de traiter des documents "mal formés", en particulier pour le HTML.

3.6. A propos du script mis en place

Le notebook "Script_encodage_automatique.ipynb" contient plusieurs instructions conditionnelles complexes basées sur le contenu de la clé "comment" des données contenu dans les fichiers Json. . Il prend en compte diverses annotations dans les données, représentées par la clé "comment", pour déterminer comment les éléments XML doivent être organisés et imbriqués. Il crée une série d'éléments XML en utilisant la bibliothèque `etree.ElementTree` (importée sous le nom `ET`). Par exemple, des éléments tels que "TEI", "body", "teiHeader", "filedesc", "div", "u", "quote", "seg", "incident" ou "table" sont créés. Les balises "seg" sont spécifiquement traitées dans le script. Par exemple, lorsqu'un commentaire contient "seg", un nouvel élément "seg" est créé avec le texte associé ou encore si le commentaire contient "quote-beginning", un élément "quote" est créé et ajouté à l'élément parent actuel.

Une fois que les éléments XML ont été créés et organisés, ils sont écrits dans un fichier XML. Le script ajoute aussi des instructions spécifiques pour le traitement des XML, indiquant probablement un schéma XML pour la validation. Le script contient une section dédiée au nettoyage des fichiers XML créés. Il utilise des expressions régulières pour effectuer des remplacements spécifiques, comme la suppression des tirets suivis de retours à la ligne.

Dans les prochaines sections, je vais détailler certaines parties majeures de notre balisage, en mettant particulièrement l'accent sur les balises qui ont posé des problèmes d'ambiguïté ou qui étaient plus complexes à implémenter. Ces éléments ont été cruciaux dans notre processus et méritent une analyse approfondie pour comprendre les défis que nous avons rencontrés et les solutions que nous avons envisagées.

Métadonnées ou les éléments du `teiHeader`

Les métadonnées sont insérées dans l'élément `<teiHeader>` de chaque fichier du corpus. L'élément `<fileDesc>` renferme des informations bibliographiques sur le fichier électronique ainsi que sur sa source d'origine grâce à l'élément `<sourceDesc>`. Dans le cadre de ce projet, ce dernier est employé pour encoder les informations de la source numérisée : le Journal officiel des débats parlementaires de la Chambre des députés.

Le titre et les sous-titres du document sont contenus dans l'élément `<titleStmt>`, qui comprend également des détails sur la réunion ou conférence via `<meeting>`, ainsi que sur les responsabilités et financements. Les titres sont fournis en français et en anglais grâce à l'attribut XML-`lang`. Le titre principal de ces documents est "Journal officiel de la République française. Débats parlementaires", avec le sous-titre "Chambre des députés : compte rendu in-extenso".

L'élément `<meeting>` est d'une importance particulière puisqu'il est utilisé pour définir le type de session (ordinaire ou extraordinaire), le numéro de la législature, et le numéro de la séance. Ces détails sont essentiels pour contextualiser le contenu des débats parlementaires.

Je n'entrerais pas en profondeur concernant cette section du processus. Cependant, une fonction spécifique a été développée¹⁷ pour gérer cette partie. Si le lecteur souhaite une compréhension détaillée, je l'invite à consulter directement cette portion du script.

```
def create_teiheader():
    """ cette fonction crée le teiHeader """

    # fileDesc
    fileDesc = ET.SubElement(teiHeader, "fileDesc")

    # Titre
    global titleStmt
    titleStmt = ET.SubElement(fileDesc, "titleStmt")
    title_fr = ET.SubElement(titleStmt, "title", type="main")
    title_fr.attrib["{http://www.w3.org/XML/1998/namespace}lang"] = "fr"
    title_fr.text = "Journal officiel de la République française. Débats parlementaires"
    title_en = ET.SubElement(titleStmt, "title", type="main")
    title_en.attrib["{http://www.w3.org/XML/1998/namespace}lang"] = "en"
    title_en.text = "Official Journal of the French Republic. Parliamentary debates"
```

Fig. 6 : Extrait de la fonction Create_teiHeader

3.7. Balisage physique

Le balisage physique fait référence à la manière dont la structure d'un document est encodée. Cette structure est généralement divisée en deux catégories distinctes :

- Les éléments structurels formels, qui concernent la mise en forme physique et la présentation du document.
- Les éléments structurels logiques, qui se rapportent à la manière dont le contenu est organisé sur le plan du sens et de la logique.

Ainsi, tandis que les éléments formels peuvent concerner des aspects comme la mise en page ou le style typographique, les éléments logiques traitent de la structuration du contenu, telle que les titres, sous-titres, ou listes, selon leur importance et leur relation les uns avec les autres.

1. Balisage formel

Le balisage formel est lié aux caractéristiques éditoriales d'un document, notamment la mise en page et la typographie spécifiques à la publication, comme celles du Journal Officiel.

¹⁷ Voir la fonction create_teiheader() disponible ici : https://github.com/mpuren/agoda/tree/stage_AGODA_2023/Script_encodage_version_finale

Même si ce balisage est déterminant, il ne représente pas l'essentiel de notre méthode d'encodage. Certains aspects sont exclus de notre balisage¹⁸.

Cependant, certains éléments formels sont conservés dans notre méthode d'encodage. Un de ces éléments est le "changement de page".

Le changement de page

L'encodage des changements de page permet de simplifier la référence à la source originale. Cette information est marquée par deux balises différentes en fonction du contexte d'utilisation.

- Lorsque le changement de page ne se produit pas dans une balise <incident>, la balise autofermante <pb> est utilisée. Conformément aux directives de la TEI, ce balisage n'est pas autorisé à l'intérieur de la balise <incident>. L'attribut "n" de la balise <pb> est utilisé pour indiquer le numéro de page.
- Si le changement de page survient à l'intérieur de la balise <incident>, alors la balise autofermante <ref> est utilisée. Cette balise est complétée par l'attribut "target", qui spécifie également le numéro de la page en question.

```
fichier_json = os.path.join(dossier_json, fichier_json)
filename = os.path.basename(fichier_json)
page_number = filename.split("_p0")[1].split(".")[0]

bp_element = ET.Element("pb", attrib={"n": "{}".format(page_number)})
# bp_element permet de récupérer le numéro de page et de l'afficher sous forme <pb n="xxx"/>
bp_element.addprevious(write_comment(filename)) # ajoute un commentaire avant le <pb n="xxx"/>
```

Fig. 7 : Extrait du code

La ligne `page_number = filename.split("_p0")[1].split(".")[0]` est utilisée pour extraire le numéro de page à partir du nom du fichier. Le nom du fichier tel que défini par l'ODD contient une structure spécifique où "_p0" est suivi du numéro de page. La ligne `bp_element = ET.Element("pb", attrib={"n": "{}".format(page_number)})` crée un nouvel élément XML nommé "pb". Cet élément a un attribut "n" qui contient le numéro de page extrait précédemment. Cela générera un élément comme `<pb n="175"/>`. La ligne `bp_element.addprevious(write_comment(filename))` ajoute un commentaire XML juste avant l'élément "pb".

¹⁸ Il s'agit des aspects liés à la mise en page tels que les alinéas, les sauts de ligne, les changements de colonne, et l'emplacement général du texte ; des détails typographiques, par exemple, l'utilisation de caractères gras ou de petites capitales, ne sont pas non plus pris en compte ainsi que certains aspects décrits dans le mémoire de Fanny Lebreton.

La principale difficulté évoquée ici concerne l'intégration de l'élément "pb" (pour "page break" ou changement de page) dans l'arborescence du document XML. Lorsque des documents sont numérisés et transformés en fichiers JSON, chaque fichier représente une page du Journal officiel. Cependant, un changement de page peut se produire à tout moment, par exemple au milieu d'un paragraphe ou lors d'une prise de parole.

Les commentaires (comment) dans les données renseignent sur la manière dont un paragraphe doit être balisé et dans quelle balise parent il doit être placé. Cette information rend difficile la prévision de l'endroit où un changement de page pourrait se produire.

Pour remédier à cette complexité, une approche a été mise en place. Elle consiste à insérer les balises de changement de page dans les éléments parents en cours de traitement. Pour ce faire, une variable "current parent" a été initialisée au début du script avec une valeur nulle. Cette variable est mise à jour à mesure que le traitement progresse, reflétant l'élément parent en cours de traitement. Si, par exemple, l'élément parent en cours est une prise de parole (<u>), et si le commentaire indique un changement de page, alors la balise de changement de page serait ajoutée au <u>.

Cependant, il a été observé que pour certaines raisons non spécifiées, l'insertion des changements de page ne se fait pas correctement lorsqu'ils surviennent au niveau des éléments "segs".

```

if 'page-number' in data[i]['comment'] and "-ref" not in data[i]['comment']:

    if current_parent is u_element:
        # Ajouter les balises 'seg' à l'élément 'u'
        u_element.append(bp_element)

    elif current_parent is quote:
        # Ajouter les balises 'seg' à l'élément 'quote'
        quote.append(bp_element)

    elif current_parent is quote_seg_beg:
        # Ajouter les balises 'seg' à l'élément 'quote'
        quote_seg_beg.append(bp_element)

    elif current_parent is note_voterlist:
        # Ajouter les balises 'seg' à l'élément 'voterslist'
        note_voterlist.append(bp_element)

    elif current_parent is comment_note :
        comment_note.append(bp_element)

    elif current_parent is comment_beg_note:
        comment_beg_note.append(bp_element)

    else :
        for div_cible in divs_cibles:
            div_cible.append(bp_element)

if data[i]['comment'] == "page-number-ref" :
    page_incident = ET.SubElement(div_sitting, "incident")
    page_inc_desc = ET.SubElement(page_incident, "desc")
    page_inc_desc.text = '''<ref target="#p''' + page_number + '''>'''

```

Fig. 8 : Extrait du code

- **Paragrophes**

Un autre élément que nous avons choisi de reproduire concerne les paragraphes, qu'ils soient dans le texte principal ou dans les sections complémentaires. Pour cela, nous utilisons la balise <seg> afin d'offrir une meilleure lisibilité au contenu. Cette balise devrait normalement inclure l'attribut XML:id, lequel donne la possibilité d'attribuer à chaque paragraphe un identifiant unique, facilitant ainsi leur repérage.

Selon les guidelines de la TEI¹⁹, l'élément "seg" en XML est utilisé selon le jugement de l'encodeur pour marquer n'importe quelle partie du texte jugée pertinente pour un traitement numérique. Il sert notamment à baliser des traits textuels pour lesquels aucun autre balisage spécifique n'est prévu. De plus, cet élément peut être utilisé pour attribuer un identifiant à un segment de texte qui est référencé par un autre élément, fonctionnant ainsi comme une ancre ou une cible pour un élément "ptr" ou un élément équivalent.

¹⁹ <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-seg.html>

Dans l'ensemble du processus d'implémentation du script, la manipulation des éléments "seg" s'est avérée être le défi le plus ardu. Même maintenant, alors que je rédige ce mémoire, je continue de reconsidérer et de peaufiner le script. Durant l'étape d'océrisation et lors de l'annotation des commentaires, les paragraphes - ou plutôt les phrases - qui doivent être encapsulés dans la balise XML <seg> peuvent avoir diverses représentations ou interprétations :

- **Seg avec « incident » :**

Lors de l'encodage, un type spécifique de "seg" correspond à la description d'un incident : il s'agit de textes qui décrivent des événements perturbant le débat, tels que des bruits, cris ou protestations. Ces descriptions sont typiquement encapsulées entre parenthèses et présentées en italique dans le texte, comme "(bruits)", "(cris)", ou "(applaudissements à droite)". Dans l'implémentation du script, j'ai intégré une fonction pour identifier ces parenthèses et insérer les balises appropriées. Cependant, un défi majeur est survenu : un seul paragraphe peut contenir plusieurs de ces incidents. Le défi réside alors dans la capacité de l'algorithme à continuer d'identifier et de baliser les incidents subséquents dans le même paragraphe après avoir traité le premier.

- **Seg interrompu par une colonne ou un changement de page :**

L'élément "seg" peut également être interrompu par une transition de colonne ou de page. Dans ces cas, le segment qui commence à la fin d'une colonne et se poursuit au début de la colonne ou de la page suivante est identifié par "seg-beginning", tandis que la fin de ce paragraphe interrompu est marquée par "seg-end". Idéalement, tout texte identifié par "seg-end" devrait correspondre et être ajouté à un "seg-beginning" précédent. Toutefois, dans la réalité, cette correspondance ne se produit pas systématiquement, ce qui ajoute une couche de complexité à la tâche de balisage automatique.

En effet, lors de l'exécution du code, nous pouvons rencontrer des situations complexes où des commentaires, tels que "seg-beginning" couplé à un incident, sont interrompus par un changement de colonne ou de page. Dans la colonne suivante, un "seg-end" est identifié, et il est logique de penser que ce "seg-end" doit être associé au "seg-beginning" avec l'incident. Toutefois, cette association n'est pas évidente pour un algorithme.

La complexité réside dans le fait qu'en algorithmique, il est difficile de traduire des règles basées sur le contexte humain, en particulier lorsque le système de commentaire offre une variété de clés qui peuvent être combinées de multiples manières. Par exemple, à part notre cas "seg-beginning" avec incident, nous pourrions avoir "seg-beginning quote" ou "seg-beginning u-beginning", entre autres.

Chaque variation nécessite une considération spéciale, ce qui rend la tâche de balisage algorithmique très délicate. L'algorithme doit être capable de reconnaître non seulement les différents types de balises, mais aussi de comprendre le contexte dans lequel elles apparaissent

pour les associer correctement. La multiplicité des combinaisons possibles de clés de commentaire complique davantage la création d'un algorithme universellement efficace pour toutes les situations.

L'encodage XML nécessite une structure hiérarchique claire où chaque élément et sous-élément sont correctement imbriqués les uns dans les autres. La difficulté dans la mise en œuvre du script réside dans la manière dont cette hiérarchie est gérée, surtout lorsqu'il s'agit d'encodages complexes avec de multiples niveaux d'imbrication.

Par exemple, avant même de baliser un segment particulier `seg`, plusieurs éléments parents doivent d'abord être créés, à savoir : l'élément racine `<TEI>`, le corps du document `<body>`, la division représentant la séance `<div type="sitting">`, une sous-division `<div type="part">`, un élément représentant une prise de parole `<u>`, et enfin le `<seg>` en question.

Ce qui complique davantage les choses, ce sont les commentaires contenant le suffixe `"-end"` qui déterminent la fin d'un élément parent. Si le script ne gère pas correctement ces commentaires, il peut y avoir des incohérences dans l'encodage. Une prise de parole mal encodée ou terminée prématurément, par exemple, pourrait entraîner la perte d'informations ou d'annotations importantes, comme le contenu exact de ce que le député a dit.

Imaginons qu'un député s'exprime. Nous initions alors une balise `<u>` pour marquer le début de son intervention, suivie de balises `<seg>` pour coder ses paroles. Si ce député se met à lire une lettre, nous entrons dans une séquence de citation marquée par la balise `<quote>`. À ce moment, le balisage `<seg>` est interrompu. Lorsque nous clôturons la balise de citation, la séquence `<seg>` n'est pas systématiquement reprise. Bien que mon script offre des résultats convaincants à ce niveau, quelques `<seg>` échappent occasionnellement à l'encodage. Dans les fichiers XML actuels, il arrive que certaines balises `<seg>` manquent juste après un `</quote>`. Dans d'autres cas, si pour une raison quelconque l'algorithme ne parvient pas à identifier la fin correcte de cette prise de parole (à l'aide d'un commentaire `"u-end"`), certains segments (`seg`) qui devraient être inclus à l'intérieur de cette balise peuvent être omis ou mal placés.

Ces complications mettent en évidence l'importance d'une gestion minutieuse de la hiérarchie et de l'imbrication lors de l'encodage XML. Pour garantir la précision et la fiabilité de l'encodage, il est crucial de tester en profondeur le script et de prévoir des exceptions ou des cas particuliers pour gérer toutes les situations possibles.

Abordons encore une autre complexité relative aux `<seg>` avec les suffixes « `beginning` » et `"-end"`. Un `"seg-end"` peut correspondre à plusieurs types de `"seg-beginning"`, tels que `"u-beginning seg-beginning"`, `"u-beginning seg-beginning incident"`, `"seg-beginning quote"`, ou `"seg-beginning incident"`. La variabilité de ces variables en fonction des commentaires rend l'obtention de résultats fiables compliquée. Si, dans le script, nous établissons la règle stipulant que tous les `"seg-end"` doivent correspondre à un `"seg-beginning"`, alors dans les situations où

un "seg-beginning incident" est suivi par un commentaire "seg-end", il est évident que ce "seg-end" se liera à un élément "seg-beginning" plutôt qu'à un "seg-beginning incident". Cette observation est applicable à tous les types de segs. Pour illustrer mes propos, considérons l'exemple suivant :

```
</seg>
<seg>Voilà comment les apôtres du parti républicain, dans les
temps héroïques, à l'époque où vous luttiez non pas pour vous
partager le pouvoir et les portefeuilles, mais</seg>
▼<seg>
M. Cuneo d'Ornano. Je ne sais quel est celui de mes collègues
qui m'a interrompu. Je le prierai de réserver son interruption
pour tout à l'heure, lorsque l'honorable rapporteur M. le comte
Lemerrier montera à cette tribune et pourra répondre avec plus
de compétence que moi, lui qui, au lendemain de la réunion des
commissions mixtes, a eu l'honneur d'être candidat officiel de
S. M. l'empereur Napoléon III. pour la doctrine du parti
républicain, pour ses principes, pour son idéal, voilà comment
les apôtres du parti républicain flétrissaient les hautes
cours de justice !
▼<incident>
<desc>(Rires à droite. – Applaudissements ironiques sur
quelques bancs à l'extrémité gauche de la salle.)</desc>
</incident>
</seg>
</u>
```

Sur la gauche, on voit le résultat généré par mon script, tandis que sur la droite, on observe le résultat attendu. Le texte en bleu est associé au commentaire « seg-beginning » et le texte en orange est lié au commentaire « u-end seg-end ». Voici la situation : le commentaire "u-end seg-end" a été lié à un texte ayant le commentaire "u-beginning seg-beginning" en raison des règles que j'avais définies pour relier ces deux commentaires. Dans mon code, chaque balisage est lié à une variable spécifique. Cependant, obtenir un résultat précis est un défi car, dans certains cas, un "seg-beginning" pourrait être suivi d'un commentaire comme "seg-end quote". Cette variabilité et combinaison de commentaires rendent la gestion de cette section du code particulièrement ardue.

Dans des conditions normales, le "u-end seg-end" devrait s'accoupler au "seg-beginning" qui le précède. J'ai tenté de mettre en place une vérification pour assurer qu'un "seg-end" est bien précédé d'un "seg-beginning", mais cette démarche s'est avérée trop complexe. De plus, le texte en noir sur blanc, situé entre les deux couleurs, est mal placé : il aurait dû être positionné bien plus haut. Lors de l'exécution du script, le "u-end seg-end" a provoqué ce déplacement indésirable du texte.

À travers ces exemples, il est clair que l'ajout successif de tags dans un commentaire "comment" facilite le processus de balisage. Cependant, cette méthode peut également compliquer la tâche si les commentaires associés ne sont pas définis avec une précision équivalente.

Il est ardu d'exposer l'entière complexité du script dans ce mémoire, notamment à cause des restrictions liées aux consignes données pour le rendu du mémoire. Je vais me limiter à ces exemples pour le moment et synthétiser les éléments qui rendent la généralisation du script

difficile. Aller plus loin reviendrait à s'égarer dans des détails que seuls ceux profondément impliqués dans le projet peuvent pleinement appréhender.

Pour finir avec cette partie, nous retrouvons la même logique d'ajout à des éléments parents. Si l'élément parent actif est, par exemple, un `u_element`, une quote, ou un commentaire, le `<seg>` y est intégré. Cet élément parent est par la suite intégré à la division en cours. Cette dernière est à son tour intégrée soit à la division de la séance en cours, soit à la division `other_sitting` si deux séances se sont déroulées le même jour. La variabilité et l'interchangeabilité de ces éléments ajoutent une couche supplémentaire de complexité au script. Les clés, sur lesquelles notre script se repose, ne sont pas fixes et peuvent changer, rendant ainsi la tâche d'automatisation encore plus délicate.

```
if data[i]["comment"] == "seg":
    # Initialisation de l'élément seg
    seg = ET.Element("seg")
    seg.text = data[i]['text_ocr']

    # Vérifier si l'élément précédent est "quote-end"
    if i > 0 and "comment" in data[i-1] and ("quote-end" in data[i-1]["comment"] or "seg incident" in data[i-1]["comment"]):
        # Traitez l'élément seg ici
        if data[i]['text_ocr'] not in added_segs:
            added_segs.add(data[i]['text_ocr'])
    else:
        # La logique précédente pour d'autres cas
        conditions = [
            i > 0 and "comment" in data[i-1] and ("quote-end" in data[i-1]["comment"] or "table" in data[i-1]["comment"]),
            i > 0 and "comment" in data[i-1] and ("seg incident" in data[i-1]["comment"] or "seg quote incid")
        ]

        for condition in conditions:
            if condition:
                while i < len(data) - 1 and "comment" in data[i+1] and "u-end" not in data[i+1]["comment"]:
                    i += 1
                    if data[i+1]["comment"] == "seg" and data[i+1]['text_ocr'] not in added_segs:
                        seg = ET.Element("seg")
                        seg.text = data[i+1]['text_ocr']
                        added_segs.add(data[i+1]['text_ocr'])
                    break

    if current_parent is u_element:
        # Ajouter les balises 'seg' à l'élément 'u'
        if u_element is not None:
            u_element.append(seg)
```

- **Signature**

Un autre aspect que nous avons pris en compte dans notre encodage est la signature, généralement située à la fin de chaque séance, avant les sections complémentaires comme les annexes, les erratums, etc. Pour cela, nous utilisons la balise `<signed>`. Il est important de souligner que nous considérons la signature comme une section distincte et autonome du document. Contrairement à d'autres éléments, cette balise ne comporte pas l'attribut `xml:id`.

```
if data[i]['comment'] == 'signed seg back':
    # Ajouter un élément 'signed seg back' avec le contenu de la clé 'text_ocr'

    signed = ET.Element("signed")
    signed_seg_back = ET.SubElement(signed, "seg")
    signed_seg_back.text = data[i]['text_ocr']
    div_cible.append(signed)
```

Fig. 9 : Extrait du code pour les « signed »

```
<signed>
  <seg>Le Chef du service sténographique de la Chambre des députés, Emile Grosselin</seg>
</signed>
</div>
```

Résultat obtenu

2. Balisage logique

Les éléments structurels logiques définissent la manière dont le contenu textuel est organisé, englobant les éléments tels que le sommaire, les différentes sections, les titres, les tableaux, les annexes, etc. Chaque compte rendu de séance comprend systématiquement un sommaire, un corps principal de texte et des sections complémentaires.

2.1. Éléments structurels du sommaire

Le sommaire offre une vue d'ensemble des différents éléments de la séance. Chaque compte rendu contient un sommaire structuré de la manière suivante :

- Tout d'abord, le sommaire est inséré dans une subdivision représentée par la balise TEI <div>. Cette balise possède un attribut "type" dont la valeur est "contents".
- À l'intérieur de cette subdivision, nous trouvons :
 - Un en-tête représenté par la balise TEI <head>.
 - Une liste, balisée par <list>, qui comprend une série d'éléments ou d'items (représentés par la balise TEI <item>). Chaque item évoque les différentes sections de la séance. Pour faciliter leur identification, chaque item est muni d'un identifiant unique, indiqué par l'attribut TEI XML:id. (La méthode exacte pour déterminer ces valeurs d'ID reste à définir).

```
elif re.search(r"\b(<!--)sitting\b", data[i]["comment"]) and re.search(r"contents", data[i]["comment"]):
    div_content = ET.SubElement(div_sitting, "div", attrib={"type": "contents"})
    div_content.addprevious(etree.Comment("SOMMAIRE"))
    list_item = ET.SubElement(div_content, "list")
```

```
# Items

elif re.search(r"\bitem(?!-)\b", data[i]["comment"]):
    item = ET.SubElement(list_item, "item")
    item.text = data[i]["text_ocr"]

elif re.search(r"item-list", data[i]["comment"]):
    item = ET.SubElement(list_item, "item")
    item.text = data[i]["text_ocr"]
```

Fig. 10 : Extrait du code

2.2. Éléments structurels du corps du texte

Le corps du texte est la section qui développe en détail chacune des parties, incluant les interventions et discussions qui ont eu lieu lors des séances.

Ce corps se positionne immédiatement après le sommaire et s'étend jusqu'à la signature du Chef du service sténographique. Il est structuré par les éléments suivants :

- **Sous-divisions**

Chaque section spécifique du corps du texte est encodée en utilisant la balise TEI <div>, complétée par un attribut "type" dont la valeur est "part". Cette spécification permet d'indiquer que la portion de texte encadrée représente une section ou partie distincte.

D'après l'ODD, si la section a été préalablement évoquée dans le sommaire, l'élément <div> est alors enrichi de l'attribut "corresp". Cet attribut établit un lien entre la section actuelle et l'item correspondant dans le sommaire, en utilisant l'identifiant <XML:id> de cet item. En d'autres termes, cela crée une correspondance directe entre la section détaillée dans le corps du texte et sa mention préliminaire dans le sommaire. N'ayant pas pu établir des identifiants stables pour chaque balise, cette section n'a pas été élaborée dans le script. De plus, toutes les diverses valeurs associées à la clé « comment » sont correctement traitées. Par exemple, si la valeur de « comment » est « part head » ou « head part », une division est instaurée avec l'attribut « part ». Cette division est par la suite intégrée au « div sitting », représentant la division principale de la séance.

Pour chaque élément de la liste data, le script évalue le contenu de sa clé "comment" pour déterminer la structure appropriée à générer. Selon cette évaluation, un nouvel élément XML est créé, avec des attributs et des sous-éléments appropriés pour représenter des sections telles que "Part Head", "Agenda", "Appendices", "Erratum", et bien d'autres. Ces éléments sont imbriqués hiérarchiquement, en utilisant des éléments de référence comme div_sitting, back, et body. Après chaque création, les éléments sont ajoutés à une liste appelée divs_cibles.

```
if "comment" in data[i]:

    # _____Grandes divisions_____

    if data[i]['comment'] == 'part head' or data[i]['comment'] == "head part" :

        # Ajouter un élément 'part head' avec le contenu de la clé 'text_ocr'
        div_part = ET.SubElement(div_sitting, "div", attrib={"type": "part", "corresp": "#pv"})
        part_head = ET.SubElement(div_part, "head" )
        part_head.text = data[i]['text_ocr']
        div_part.addprevious(etree.Comment(generate_id("Partie_"))) # ajout de commentaire avant chaque partie suivi d'un id
        divs_cibles.append(div_part)

    elif re.search(r"part1(?:-)", data[i]["comment"]):

        div_part1 = ET.SubElement(back, "div", attrib={"type": "part"})
        div_part1.addprevious(etree.Comment(generate_id("div_part")))
        divs_cibles.append(div_part1)

    elif re.search(r"agenda", data[i]["comment"]):

        # Ajouter un élément 'agenda' avec le contenu de la clé 'text_ocr'
        div_agenda = ET.SubElement(div_sitting, "div", type="agenda")
        div_agenda.addprevious(etree.Comment("Partie_Agenda"))
        divs_cibles.append(div_agenda)
        agenda_head = ET.SubElement(div_agenda, "head")
        agenda_head.text = data[i]['text_ocr']
```

Fig. 11 : Extrait du code

- Titres

Des fois, des sections spécifiques peuvent débuter par un titre. Pour représenter ces titres, nous utilisons la balise TEI <head>. Cette balise permet d'indiquer clairement le début et le sujet d'une section ou d'une subdivision du texte. Il se peut qu'un titre soit articulé en deux segments distincts. Dans une telle situation, l'élément <head> est structuré de manière à contenir la première moitié du titre dans la balise <label> et la seconde moitié dans la balise <note>.

- Notes Structurelles

Parfois, les sténographes introduisent des informations qui ne sont ni des segments de discours, ni des commentaires de nature sémantique. Pour représenter ces informations spécifiques, nous utilisons la balise <note>. Chaque <note> est agrémentée de l'attribut XML:id, attribuant ainsi un identifiant unique à chaque information, qui la distingue et la numérote.

```
# _____ Commentaires et notes _____

elif data[i]['comment'] == 'note-head':
    # Ajouter un élément 'note-head' avec le contenu de la clé 'text_ocr'
    note_head = ET.Element("note")
    note_head.text = data[i]['text_ocr']
    # Ajouter Les éléments <note-head> à chaque div cible
    div_cible.append(note_head)

if re.search(r"voterslist-beginning", data[i]["comment"]):
    note_voterlist = ET.SubElement(div_voting, "note", attrib={"type": "voterslist"})
    voterlist = ET.SubElement(note_voterlist, "desc")
    voterlist.text = data[i]['text_ocr']

if data[i]['comment'] == 'comment seg' or data[i]['comment'] == "seg comment" or data[i]['comment'] == 'comment' :
    # Ajouter un élément 'comment seg' avec le contenu de la clé 'text_ocr'
    comment_note = ET.Element("note", attrib={"type": "comment"})
    comment_seg = ET.SubElement(comment_note, "seg")
    comment_seg.text = data[i]['text_ocr']
    div_cible.append(comment_note)
    if comment_note is not None:
        comment_note.tail = " "

if data[i]['comment'] == 'comment-beginning seg':
    # Ajouter un élément 'comment-beginning seg' avec le contenu de la clé 'text_ocr'
    comment_beg_note = ET.Element("note", attrib={"type": "comment"})
    comment_beginning_seg = ET.SubElement(comment_beg_note, "seg")
    comment_beginning_seg.text = data[i]['text_ocr']
    div_cible.append(comment_beg_note)

if data[i]['comment'] == 'comment-end seg':
    # Ajouter un élément 'comment-end seg' avec le contenu de la clé 'text_ocr'
    comment_end_seg = ET.Element("seg")
    comment_end_seg.text = data[i]['text_ocr']
    comment_beg_note.append(comment_end_seg)

if comment_note is not None:
    comment_note.tail = " "
#comment_note = ET.Element("note", attrib={"type": "comment"})
```

Fig. 12 : Extrait du code

• Tableaux

Le corps du texte peut, à l'occasion, intégrer des informations sous forme tabulaire. Pour encadrer ces données, nous utilisons l'élément TEI <table>. L'élément principal, <table>, est enrichi des attributs "rows" et "cols" qui indiquent respectivement le nombre de lignes et de colonnes du tableau.

De plus, l'élément <cell> peut être assorti de l'attribut "role", afin de clarifier le type de contenu ou la fonction de la cellule au sein du tableau.

Dans les cas où le tableau fait état des résultats d'un vote, un attribut "corresp" est ajouté à la balise <table>. Cet attribut sert à créer un lien vers l'identifiant unique du vote concerné, généralement listé dans les annexes. Automatiser ce processus s'avère complexe. Il est impératif, tout d'abord, de permettre à l'algorithme de discerner le contexte dans lequel il opère, c'est-à-dire de déterminer si ce tableau est situé dans une division de type "voting", "part" ou autre. Une fois ce repérage effectué, le défi réside ensuite dans la création d'une liaison avec l'identifiant unique de l'élément concerné. En raison de la sophistication du code, je n'ai pas été en mesure d'accomplir cette tâche.

```
if data[i]['comment'] == 'table':
    # Ajouter un élément 'table' avec le contenu de la clé 'text_ocr'
    table = ET.Element("table")
    row = ET.SubElement(table, "row")
    cell = ET.SubElement(row, "cell")
    cell.text = data[i]['text_ocr']

    if current_parent is u_element:
        # Ajouter les balises 'seg' à l'élément 'u'
        if u_element is not None:
            u_element.append(table)

    elif current_parent is quote:
        # Ajouter les balises 'seg' à l'élément 'quote'
        quote.append(table)

    elif current_parent is note_voterlist:
        # Ajouter les balises 'seg' à l'élément 'voterslist'
        note_voterlist.append(table)

    else :
        div_cible.append(table)
```

Fig. 13 : Extrait du code

Ici, nous sommes confrontés aux mêmes enjeux mentionnés précédemment concernant la variabilité des éléments parents. Une table peut être insérée dans différentes structures : une énonciation <u>, une citation <quote> ou encore une note <note>. J'ai pu identifier ces cas variés grâce à un autre script que j'ai conçu, ayant pour but de déceler les différents types de commentaires présents dans un fichier JSON. Étant donné la complexité de prédire tous les cas envisageables, j'ai opté pour une approche pragmatique. Si la table ne se trouve pas dans une citation, une note ou une énonciation, j'ai intégré une condition "else" qui l'insère directement dans la division en cours.

- **Éléments structurels des parties complémentaires**

Au-delà du corps principal du texte, les comptes rendus de débats intègrent souvent des sections supplémentaires. Ces ajouts peuvent prendre la forme d'annexes, d'erratum, de pétitions, de nominations pour divers postes, de listes, entre autres. Pour englober ces éléments, nous utilisons la balise TEI <back>.

Chaque section complémentaire est logée dans une division (<div>), et le type de cette division est spécifié selon le contenu ou le sujet traité.

- **Annexes**

Les annexes servent souvent à détailler les votes effectués lors d'une séance. Parfois, elles apportent des corrections relatives aux votes des séances antérieures. Voici comment ces annexes sont structurées :

Scrutins de la Séance

L'intégralité des scrutins est regroupée sous une division dotée de l'attribut XML:id. Pour définir la valeur de cet identifiant, nous utilisons le préfixe "CR", suivi de la date de la séance formatée selon la norme ISO 8601, et terminons par "vot".

Chaque vote individuel est lui aussi encapsulé dans une division. Cette balise <div> est renseignée par plusieurs attributs :

- **XML:id** : Sa valeur est conçue en reprenant la logique de l'identifiant de la division parente (qui contient tous les scrutins), mais s'ajoute à cela un numéro pour distinguer chaque scrutin.
- **type** : Cet attribut a pour valeur "voting", signifiant qu'il s'agit d'un scrutin ou d'un vote.
- **corresp** : Il sert à établir un lien entre le scrutin actuel et le sujet de ce vote, en renvoyant précisément à l'identifiant <XML:id> de l'élément correspondant listé dans le sommaire.

3.8. Balisage sémantique

Le balisage sémantique constitue une étape importante dans l'encodage de textes. Il ne se contente pas de reconnaître et de structurer des éléments physiques ou logiques dans un document, mais il cherche à identifier et à attribuer une signification ou une interprétation à des mots ou des expressions précis (Verlaet, 2010). Ce type de balisage donne une dimension analytique au contenu, ce qui permet d'extraire des informations plus significatives et pertinentes.

Deux grands domaines sont couverts par le balisage sémantique :

1. Éléments Constitutifs du Discours :

- **Prises de parole** : Ces éléments identifient les différentes interventions ou déclarations d'un individu ou d'un groupe dans le texte. Ils peuvent être marqués, par exemple, pour distinguer les déclarations d'un président de celles d'un membre de l'audience.
- **Commentaires** : Ce sont les remarques ou explications insérées dans le texte, souvent pour donner un contexte, préciser une idée ou apporter une nuance.
- **Citations** : Elles repèrent les extraits empruntés à d'autres sources, qu'il s'agisse de discours antérieurs, d'œuvres écrites, de lois ou de toute autre référence.

2. Entités Nommées :

Il s'agit d'informations spécifiques qui sont identifiées et classées selon leur nature. Les entités nommées comprennent :

- **Personnes** : Il peut s'agir de noms propres, de titres ou de tout autre identifiant associé à une personne spécifique.

- **Lieux** : Cela englobe non seulement les noms géographiques (villes, pays, rivières, montagnes, etc.) mais aussi des lieux spécifiques comme des bâtiments ou des salles.
- **Organisations** : Cela inclut les noms d'entreprises, d'institutions, d'agences gouvernementales, de groupes, etc.
- **Éléments Temporels** : Cela peut aller des dates spécifiques aux jours de la semaine, en passant par les périodes historiques ou les événements récurrents.
- **Quantités** : Ces éléments peuvent englober les chiffres, les mesures, les pourcentages, et toute autre information quantifiable.

En fin de compte, le balisage sémantique sert à enrichir le texte, en transformant un simple contenu en une source d'informations structurées et facilement accessibles. Cela permet non seulement d'améliorer la compréhension du texte, mais aussi de faciliter la recherche et l'extraction d'informations spécifiques. Je me concentrerai uniquement sur le premier aspect, celui concernant les éléments fondamentaux du discours.

1. Éléments du discours

1.1. Énoncés

L'énoncé (Utterance) est un élément capital dans la transcription de discours ou de débats, car il permet d'attribuer des propos spécifiques à un orateur donné. Il sert de repère pour comprendre qui s'exprime à un moment donné et quelle est sa contribution à la conversation ou au débat. Il sert d'indicateur signalant un changement de locuteur ou une transition dans le discours.

L'indication de l'orateur au début de l'énoncé est essentielle pour délimiter et attribuer clairement les prises de parole. Cela permet aux lecteurs ou aux chercheurs de suivre le flux de la discussion et de comprendre les différents points de vue présentés.

L'utilisation de l'élément <u> de la Text Encoding Initiative (TEI) est particulièrement adaptée à cette tâche. En effet, l'élément <u> est spécifiquement conçu pour marquer une unité de discours, que ce soit une phrase, un paragraphe, ou une prise de parole complète.

À l'intérieur d'un énoncé, il peut y avoir une structure plus détaillée pour délimiter les différentes parties du discours. Cela peut inclure des paragraphes individuels, des commentaires intercalaires ou d'autres types d'annotations.

L'utilisation de l'élément <seg> est appropriée pour marquer des segments ou des paragraphes distincts à l'intérieur de l'énoncé. Quant aux commentaires, ils peuvent être inclus à l'aide de

l'élément <note>, qui est couramment utilisé pour ajouter des annotations ou des remarques explicatives dans un texte encodé TEI.

Il y a trois principales catégories de commentaires qui mènent au balisage <u> : ceux qui contiennent "u seg", ceux avec "u-beginning" et enfin, ceux avec "u-end".

Le commentaire "u seg" sert d'indicateur pour une intervention qui consiste en un seul paragraphe ou phrase. Lorsque le commentaire est défini comme "u seg", une balise <u> est générée, suivie par une balise enfant <seg>. Le texte OCR est ensuite intégré dans cette balise <seg>, qui est à son tour inséré dans l'élément parent <u>. À cette étape, tout se déroule sans encombre.

Cependant, nous rencontrons des défis semblables à ceux mentionnés précédemment avec les balisages <seg>, lorsqu'il s'agit de commentaires avec les suffixes "-beginning" et "-end" concernant les balisages <u>. Je n'approfondirai pas davantage sur les complexités des <seg>, sauf pour souligner que les balises <u> sont étroitement liées aux balises <seg>. La nuance est qu'un "u-beginning" déclenche l'ouverture d'une balise, qui se referme ensuite lorsqu'un commentaire "u-end" est détecté. De plus, un "u-beginning" ou "u-end" peut être couplé à d'autres commentaires, comme "incident", "quote" ou « div-end », demandant ainsi des traitements additionnels. De plus, un autre risque est l'absence de "u-end". Nous avons observé que certains « box » manquent suite au processus d'OCéRisation. L'absence de ces box et des commentaires associés, peut entraîner la génération d'un fichier XML erroné ou incomplet. Le risque réside dans le fait que certaines interventions peuvent se confondre avec celles d'autres intervenants si les "u-end" ne sont pas correctement identifiés ou mal positionnés.

```
if data[i]["comment"] == "u-beginning seg quote" :
    seg_cas = ET.Element("seg")
    text_ocr = data[i]["text_ocr"]

    # Rechercher l'indice de la première occurrence de "«" et de "»" dans le texte
    start_index = text_ocr.find("«")
    end_index = text_ocr.find("»")

    if start_index != -1 and end_index != -1 and start_index < end_index:
        # Extraire la partie du texte entre les guillemets
        quote_text = text_ocr[start_index : end_index+1]

        # Créer un élément <quote> et y ajouter le texte extrait
        quote_seg = ET.SubElement(seg_cas, "quote")
        quote_seg.text = quote_text

        # Insérer l'élément <quote> dans la bonne position en utilisant les méthodes d'insertion
        seg_cas.text = text_ocr[:start_index]
        seg_cas.insert(1, quote_seg)
        quote_seg.tail = text_ocr[end_index + 1:]

        u_element.append(seg_cas)
    else:
        # Aucun guillemet trouvé ou l'ordre est incorrect, utiliser le texte tel quel
        seg_cas.text = text_ocr
        u_element.append(seg_cas)
```

Fig. 14 Exemple d'un code traitant un commentaire « u-beginning seg quote »

1.2. Commentaires des sténographes

Les comptes rendus des débats parlementaires sont souvent dotés de commentaires contextuels. Ces notes, ajoutées par les sténographes, sont déterminantes pour fournir un contexte à la prise de parole des orateurs. Ces commentaires font office de repères temporels, signalant le début et la fin des sessions, mais aussi les éventuelles interruptions, les actions telles que les votes ou d'autres détails pertinents.

Il est essentiel de distinguer ces commentaires contextuels des simples remarques structurelles. Pour mieux organiser ces annotations, nous les classons en deux catégories principales.

Premièrement, les commentaires liés à l'organisation de la séance et aux différentes actions qui y sont menées sont balisés avec l'élément TEI <note>. L'attribut type de cet élément permet de préciser la nature du commentaire. Les valeurs possibles pour cet attribut sont :

"opening" : pour indiquer le début de la session.

"closing" : pour signaler sa conclusion.

"result" : pour annoncer les résultats des votes.

"comment" : pour toute autre précision ou information contextuelle.

Deuxièmement, pour évoquer l'ambiance ou tout événement qui perturbe le déroulement normal de la séance, nous utilisons l'élément TEI <incident>. À l'intérieur de cet élément, un sous-élément <desc> fournit une description de l'événement ou du phénomène.

Par exemple, si un orateur est interrompu par des applaudissements ou un incident technique, cet élément est utilisé pour le noter. Ces marques permettent aux lecteurs d'avoir une meilleure compréhension du déroulement de la séance et du contexte dans lequel les paroles ont été prononcées.

```

▼<seg>
  La plus importante des rectifications concerne le procès-verbal. En effet, au moment où la Chambre s'est
  prononcée en majorité en faveur de la validation d'un de nos collègues de l'Ardèche, ni mes amis ni moi
  n'avons applaudi au succès de telle ou telle candidature représentant telle ou telle opinion ; ni mes amis
  ni moi n'avons même applaudi à la défaite de ce que j'appellerai le groupe des invalideurs...
  ▼<incident>
    <desc>(Bruit à gauche.)</desc>
  </incident>
</seg>
</u>
▼<u>
  <seg>Voix au centre. Ce n'est pas une rectification.</seg>
</u>
▼<u>
  ▼<seg>
    M. Paul Déroulède. Mes amis et moi avons applaudi à ce que nous avons cru, à ce que nous croyons encore être
    la formation spontanée d'une majorité de conscience, de justice, de tolérance et d'union.
    ▼<incident>
      <desc>(Applaudissements sur quelques bancs à l'extrémité gauche de la salle. – Exclamations au centre et à
      gauche.)</desc>
    </incident>
  </seg>
</u>
- ...

```

Fig. 15 : Extrait du fichier xml

J'ai précédemment évoqué les difficultés liées aux incidents. Pour optimiser le script sur cet aspect, il serait judicieux de développer une approche ciblée pour identifier les situations où plusieurs incidents se manifestent dans un unique seg. Une stratégie envisageable serait d'intégrer des commentaires dédiés à ces situations et de fragmenter le seg en sections. Par exemple, lors de l'ajout de commentaires, on pourrait intégrer "seg incident part1" et "seg incident part2". Ainsi, un script dédié pourrait être conçu pour traiter cette section, en fusionnant les deux parties dans un seg unique pour le document XML final. Bien que cette approche simplifie le processus, elle introduit un nouveau commentaire à considérer dans le guide d'encodage.

La compréhension contextuelle est essentielle pour éviter des erreurs de balisage. Puisque le script a été programmé pour identifier un incident en se basant sur le contenu entre parenthèses, cela peut conduire à des confusions lorsque, par exemple, le prénom d'un député est également mis entre parenthèses dans le compte rendu pour préciser le député qui s'exprime (s'il y a plusieurs députés ayant le même nom). Dans ce cas, le script peut mal interpréter cette parenthèse comme l'incident, alors qu'en réalité, l'incident est situé ailleurs. Il serait alors nécessaire de développer une logique plus sophistiquée pour discerner ces nuances. L'image ci-dessous montre un exemple dans lequel (Pierre) est balisé alors que l'incident à baliser est (Rires à gauches.)

```

▼<u>
  ▼<seg>
    Ce n'est pas encore tout. Une déclaration, également légalisée par le maire et revêtue de cinq signatures,
    nous fait connaître que le sieur Bessède
    ▼<incident>
      <desc>(Pierre)</desc>
    </incident>
    , curé de Faillagol, lit en chaire, à ses paroissiens, non pas le mandement de l'évêque ni la lettre
    pastorale, mais devinez quoi? le journal l'Autorité!... (Rires à gauche.)
  </seg>
</u>

```

Fig. 16 : Extrait du fichier xml

1.3. Citations

Les orateurs, lors de leurs interventions, font souvent référence à des passages ou des expressions issus d'autres textes ou discours, pour appuyer, contredire ou éclairer leurs propos. Dans les comptes rendus, ces emprunts sont généralement indiqués entre guillemets pour les distinguer du reste du discours. Afin de normaliser l'encodage de ces références externes, la balise TEI <quote> est utilisée. Elle a été spécifiquement conçue pour baliser des segments de texte qui ont été empruntés à une autre source. Grâce à cet élément, les citations sont clairement identifiées, offrant ainsi aux lecteurs une compréhension précise de la structure du discours et de la distinction entre les paroles propres de l'orateur et les emprunts à d'autres sources.

Nous avons brièvement évoqué le balisage des citations. Il est important de différencier "quote seg" (ou "quote-seg") de "quote-beginning". Le premier fait référence à une citation déjà

encapsulée dans les segs, tandis que le second sert d'indicateur pour l'ouverture d'un élément parent <quote> destiné à contenir plusieurs éléments segs. Il est évident que la combinaison des commentaires "quote" et "incident" pose problème, car il faut identifier, au sein du même texte, les parenthèses et les guillemets qui signalent respectivement un "incident" et une "quote". L'enjeu réside dans la capacité à baliser correctement ces éléments tout en garantissant une bonne réassemblage des éléments sans créer de doublons.

3.9. Vérification de la conformité des fichiers TEI au schéma utilisé

Pour garantir l'intégrité et la qualité des documents TEI, il est essentiel de vérifier leur conformité à un schéma. L'un des avantages de LXML est sa capacité à valider des documents XML par rapport à différents types de schémas. Il prend en charge plusieurs types de schémas, comme le DTD, et XML Schéma et le schéma RNG (Relax NG). Pour vérifier la conformité avec LXML, l'utilisateur écrit du code puis charge le schéma RNG et le fichier TEI qu'il souhaite valider. On exécute le code dans une notebook pour vérifier si le fichier TEI est conforme au schéma. En cas d'erreur, LXML fournit des messages détaillés pour aider à identifier et résoudre les problèmes.

Il est également possible de faire cette vérification de conformité avec Oxygen XML Editor. En plus de ses nombreuses fonctionnalités, Oxygen XML Editor est un éditeur XML professionnel qui offre une fonctionnalité intégrée pour valider des documents XML par rapport à différents types de schémas, y compris RNG. C'est avec ce dernier que j'ai effectué la vérification des fichiers XML générés tout au long de mon projet. À la différence de LXML, Oxygen ne nécessite pas de programmation pour réaliser cette vérification. Il permet d'associer aisément un schéma RNG au fichier à examiner. Après avoir lié le schéma, l'éditeur propose une fonction de validation. Si le fichier est conforme, Oxygen confirme par un message positif « validation réussie ! ». Sinon, il présente une liste précise des anomalies, guidant l'utilisateur vers leur résolution.

4. Résultats et réflexion

4.1 Présentation des résultats obtenus lors du stage

Le script élaboré pendant mon stage a donné des résultats satisfaisants. Lors de la première validation selon le schéma défini, nous avons détecté plus de 750 erreurs dont la majorité concernait les <seg>. Ces anomalies ont conduit à des améliorations significatives du script. Après avoir généré un fichier XML qui correspondait à celui produit lors d'un stage antérieur et qui s'approchait de l'encodage idéal, j'ai élargi mes tests à deux autres séances, celles des 28 et 30 novembre. Rappelons qu'une séance se compose de multiples pages numérisées et converties en format JSON. Grâce au script élaboré, la conversion des fichiers json en un fichier XML ne se fait qu'en quelques secondes.

Les fichiers XML produits sont conformes au schéma RNG défini. Pour les trois fichiers en question, la validation a été concluante, attestant de la conformité de nos fichiers XML avec les standards requis pour le projet AGODA.

L'outil Oxygen XML Editor propose une fonctionnalité très utile qui permet de comparer deux documents. Suite à cette comparaison, à part l'enrichissement des commentaires, l'ajout d'attributs aux éléments et quelques autres améliorations mineures, l'analyse a démontré la fiabilité et l'efficacité du script. De mon point de vue, le script atteint une performance d'environ 90 à 95%. Les quelques lacunes restantes (soit 5 à 10%) peuvent être attribuées aux défis et limitations abordés tout au long de ce mémoire, notamment les imprécisions des commentaires dans les fichiers JSON, la nécessité d'une association précise entre les variables de ces commentaires dans le script, et l'omission occasionnelle de certains commentaires.

4.2 Analyse des avantages et des limites de la méthode employée

La méthode que nous avons adoptée dans le cadre de ce projet a démontré son efficacité et présente de nombreux avantages. Cependant, comme toute approche, elle comporte également ses limites.

Un des points forts de l'utilisation de la bibliothèque LXML est qu'elle permet d'obtenir un code à la fois compact et clair. En comparaison avec le code écrit par l'ancien stagiaire (comme illustré ci-dessous), le code généré à l'aide de LXML se distingue par sa simplicité et sa lisibilité.

```
def add_structure(data):
    """
    Ajout des éléments TEI "text" "body" et "back" pour chaque boîte étiquetée "body", "text", "back", "text-back"
    :param data: dictionnaire contenant l'ensemble des données issues des JSON
    """
    for i in range(len(data)):
        if "comment" in data[i]:
            if re.search(r"\bbody\b", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['<text><body>', data[i]['text_ocr']])
            # elif re.search(r"body1", data[i]["comment"]):
            # data[i]['text_ocr'] = "".join(['<text><body><pb n="1"/>'])
            elif re.search(r"\btext(?:!-)\b", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['</div></body></text>'])
            elif re.search(r"\b(?:!-)\b", data[i]["comment"]):
                data[i]['text_ocr'] = "".join([data[i]['text_ocr'], '</div></div></body><back>'])
            elif re.search(r"text-back", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['</div></div></back></text>'])
            else:
                pass
    return data
```

En haut, le code écrit par l'ex-stagiaire et en bas le code avec LXML

```
# Création de l'élément racine et le squelette du XML
root = ET.Element("TEI", xmlns="http://www.tei-c.org/ns/1.0")
root.attrib["{http://www.w3.org/XML/1998/namespace}lang"] = "fr"

teiHeader = ET.SubElement(root, "teiHeader")
text_tei = ET.SubElement(root, "text")
body = ET.SubElement(text_tei, "body")
back = ET.SubElement(text_tei, "back")
div_sitting = ET.SubElement(body, "div", attrib={"type": "sitting"})
u_element = ET.Element("u")
div_voting = ET.Element("div", attrib={"type": "voting"})
div_voting.addprevious(etree.Comment("Voting"))
```

Les deux codes ci-dessus font la même chose : Ils créent la structure du fichier xml

Fig. 17 : Extrait du code et un extrait du script de F. LEBRETON

L'apprentissage de LXML s'est avéré relativement intuitif. Après avoir consulté la documentation et effectué quelques recherches, j'ai rapidement été en mesure de produire mes premières lignes de code. La plupart du temps, il m'a suffi de réutiliser le même schéma de code en modifiant simplement le type de commentaire, l'attribut à associer, et en déterminant, en fonction du commentaire, dans quel élément parent insérer le contenu.

Cependant, la complexité est apparue lorsque le code devait gérer de multiples éléments parent. Si la tâche se résumait simplement à écrire des balises, les associer à un texte et les insérer dans un élément "body", alors le processus serait direct. Mais le défi résidait dans la nécessité de produire un document conforme à un schéma spécifique, nécessitant une approche générique tout en gérant des cas spécifiques.

Face à cette complexité, je me suis tourné vers la documentation pour trouver des solutions. C'est ici que j'ai rencontré un inconvénient majeur : la documentation sur LXML est limitée et, lorsqu'elle existe, elle est majoritairement en anglais.

Une autre limitation de LXML est sa gestion des éléments via des variables. Pour illustrer : pour imbriquer un élément "body" dans un élément parent "root", il y a des étapes précises à suivre, nécessitant la création de variables intermédiaires pour chaque élément. Cette méthode introduit une dépendance entre les variables, ce qui peut rendre la structure du code plus fragile.

```
Import lxml.etree as ET
Racine = ET.Element("root")
Corps = ET.SubElement(Racine, "body")
Ou faire :
Corps = ET.Element(Racine, "body")
Racine.append(corps)
```

Fig. 18 : Exemple de code

En revanche, la technique utilisée par Fanny Lebreton semble offrir une solution à ce problème. Sa méthode, qui consiste à pré-écrire des balises puis à les associer aux textes correspondants, évite cette dépendance aux variables. Cette méthode semble plus directe et moins susceptible aux erreurs associées à l'interdépendance des variables.

4.3 Réflexion sur l'apport du travail réalisé

L'utilisation des outils numériques dans le domaine des humanités, notamment pour l'éditorialisation et l'encodage automatique de fichiers ocrisés vers du XML, soulève de nombreuses réflexions méthodologiques. L'un des principaux défis de la conversion automatique est la fidélité du contenu. Cela signifie que les résultats de l'ocrisation (OCR) doivent être soigneusement vérifiés pour s'assurer qu'ils correspondent au document original, surtout si le document source est ancien ou si sa qualité est faible. L'utilisation du XML, et plus spécifiquement du TEI - Text Encoding Initiative, est encouragée car elle offre une structure standardisée qui facilite l'interopérabilité des données entre différents systèmes et projets. L'encodage en XML offre également la possibilité d'ajouter des métadonnées détaillées, qui peuvent être précieuses pour la recherche et l'analyse, comprenant des informations sur l'auteur, l'époque, les interlocuteurs, le lieu de publication, et plus encore.

Par ailleurs, l'importance de la mise en place d'un workflow adéquat et méthodique dans un projet d'humanités numériques, tel que celui piloté par AGODA, ne saurait être sous-estimée. La qualité des données, leur gestion et leur transformation jouent un rôle primordial dans la valorisation des contenus culturels et historiques, ainsi que dans la pérennisation des ressources numériques.

AGODA, grâce à son approche méthodique, semble avoir jeté les bases d'un tel workflow réussi. Les résultats obtenus à ce stade sont prometteurs, non seulement pour le projet en cours, mais aussi comme modèle pour d'autres initiatives similaires à l'avenir. Il est souvent dit que le succès des projets réside dans les détails, et l'étape d'océrisation en est un exemple probant. Bien qu'elle soit laborieuse, elle constitue le pilier central sur lequel repose la réussite des phases subséquentes. Un fichier OCR précis et de haute qualité simplifie considérablement le processus d'encodage. La qualité des annotations et commentaires enrichit ce fichier et, par conséquent, facilite grandement l'automatisation de l'encodage.

Néanmoins, comme dans toute entreprise technologique, il y a toujours des marges d'amélioration. Les fichiers XML générés, bien qu'ils représentent une étape significative vers l'achèvement du projet, nécessiteront des ajustements et des corrections pour répondre aux normes de publication académique ou institutionnelle.

Le script actuel, malgré ses mérites, présente des zones de répétition. Une modification est souhaitable pour le rendre plus compacte et efficace. En réduisant les répétitions et en les réorganisant en fonctions plus spécifiques, nous obtiendrons non seulement un code plus lisible mais aussi plus maintenable. Convertir ces fonctions en un fichier ".py" distinct est également une étape logique, permettant une meilleure gestion du projet et une réutilisabilité améliorée.

En conclusion, le parcours d'AGODA dans le monde des humanités numériques s'avèrent prometteur. Les défis rencontrés offrent des opportunités d'apprentissage et de croissance, et avec des ajustements et des optimisations appropriés, le projet est bien parti pour réaliser ses objectifs et servir de modèle pour d'autres initiatives. En partageant les connaissances, les outils et les meilleures pratiques, la communauté des humanités numériques peut avancer plus efficacement. Enfin, tout en reconnaissant le potentiel des outils numériques pour transformer la recherche en humanités, il est essentiel d'adopter une approche critique et réfléchie à leur utilisation.

4.4. Retour sur l'expérience personnelle du stage

Ce stage a représenté une véritable opportunité de croissance professionnelle et personnelle. Avant de me plonger dans ce projet, j'étais étranger à la librairie lxml. Grâce à cette expérience, non seulement j'ai maîtrisé cet outil, mais j'ai aussi consolidé mes compétences en XML, un langage que je connaissais mais que je n'avais pas eu l'occasion d'explorer en profondeur.

En travaillant sur le projet, j'ai constaté la flexibilité et la puissance de la librairie lxml. Cela m'a incité à envisager son utilisation dans des projets personnels à l'avenir, notamment dans la création de pages HTML. Je suis convaincu qu'adopter une approche similaire à celle utilisée lors de ce stage pourrait s'avérer bénéfique.

Ma détermination à produire un travail de qualité m'a souvent poussé à dépasser les heures conventionnelles, allant jusqu'à passer des nuits en éveil pour résoudre des problèmes complexes ou peaufiner des détails. Cette intensité a été tempérée par la liberté et l'autonomie que l'on m'a accordées. Cette confiance m'a permis de naviguer à travers les défis sans la pression constante d'une supervision étroite, ce qui m'a permis de travailler dans un environnement apaisant et propice à la créativité.

De plus, mon passage au DataLab de la BnF à Paris a été un moment marquant. C'était une occasion exceptionnelle de plonger au cœur d'une institution renommée et d'explorer les dernières innovations dans le domaine des humanités numériques. Les échanges avec les professionnels sur place m'ont offert des perspectives nouvelles et m'ont montré l'ampleur des possibilités dans le domaine des données et de la numérisation.

En somme, ce stage a été une plateforme d'apprentissage et d'expérience inestimable, me préparant à affronter des projets futurs avec une confiance et une compétence renouvelée.

5. Perspectives

5.1 Pistes d'amélioration de la chaîne de traitement

Après avoir détaillé la mise en œuvre de notre chaîne de traitement, il est pertinent de se pencher sur les aspects qui pourraient être améliorés afin d'optimiser davantage le processus. J'ai déjà évoqué plusieurs axes d'amélioration dans les sections antérieures. Dans cette partie, je me concentrerai exclusivement sur un aspect spécifique.

L'amélioration cohérente du guide d'annotation est primordiale, car c'est sur ce dernier que repose notre travail d'encodage automatique. Un guide minutieusement conçu, qui prend en compte une vaste gamme de combinaisons des annotations "comment", est crucial. Il est essentiel que ce guide soit structuré avec une précision et une cohérence rigoureuse dans la rédaction des commentaires. Par exemple, les fichiers json montrent différentes façons d'annoter qui ne sont pas toujours cohérentes avec le guide. Des annotations comme "part head" et "head part" en sont la preuve. Bien que les deux commentaires désignent principalement un titre à baliser comme "head", ils indiquent aussi qu'il doit être placé dans une division de type "part".

Dans le code, une instruction telle que : `if data[i][comment] == "head part"` est stricte et ne reconnaîtra pas une annotation comme "part head", ou "head part" avec des espaces supplémentaires, ni "head-part" avec un tiret intermédiaire. Il n'est pas non plus judicieux d'opter pour une approche trop générale comme : `if "head" in data[i][comment]`. Ce dernier risque d'interpréter incorrectement tous les commentaires contenant "head", par exemple "note head". Ainsi, la clarté et la précision sont impératives pour le guide d'annotation, garantissant une annotation cohérente et évitant des erreurs d'interprétation dans l'encodage automatique.

En matière de gestion des éléments "seg" destinés à être inclus dans la balise parent "quote", je recommande d'adopter un format de commentaire spécifique, tel que "quote-seg" ou "quote seg". Une telle uniformisation simplifierait considérablement la logique de programmation qui gère l'encodage des "seg", réduisant ainsi la complexité induite par les nombreux cas spécifiques et scénarios envisagés dans le code actuel.

De plus, il est essentiel de se concentrer sur les commentaires portant les suffixes "-beginning" et "-end". Le recours à un seul indicateur de clôture, tel que "u-end", pour divers commentaires d'ouverture, tels que "u-beginning seg", "u-beginning quote seg", "u-beginning seg incident", ou "u-beginning seg-beginning", engendre des difficultés dans l'association appropriée des

variables. Sauf si une méthode efficace est mise en place pour gérer l'ouverture et la clôture des balises avec lxml, il serait peut-être plus judicieux d'opter pour un balisage unique. Dans cette optique, il conviendrait de définir clairement comment chaque indicateur d'ouverture correspond à son indicateur de clôture, garantissant ainsi la cohérence et l'exactitude du processus d'encodage.

5.2 Recommandations pour l'automatisation des tâches avec des algorithmes entraînés

L'ajout de commentaires dans les fichiers JSON s'avère être une tâche exigeante qui requiert une précision et une uniformité méticuleuses. Dans le cadre de l'optimisation de ce processus, il est vivement recommandé d'envisager l'entraînement d'un algorithme d'apprentissage automatique pour cette mission. Utiliser l'intelligence artificielle présente de multiples avantages, tels que l'efficacité, en traitant rapidement de vastes volumes de données ; l'uniformité, en garantissant une cohérence dans l'ajout des commentaires et éliminant les variations humaines ; et l'évolutivité, permettant à l'algorithme de s'adapter à de nouveaux formats ou structures de commentaires au fil du temps. AGODA semble déjà avoir pris des initiatives dans cette direction, ce qui est prometteur. Continuer les investissements dans cette voie, notamment en privilégiant la qualité des données pour l'entraînement et en adoptant les meilleures pratiques en matière d'apprentissage automatique, serait bénéfique.

Par ailleurs, le balisage joue un rôle central dans la structuration des données. Afin d'assurer une structuration systématique et précise, l'entraînement d'un algorithme dédié à l'automatisation de cette tâche serait judicieux. Une telle démarche offre une précision accrue, permettant aux algorithmes d'identifier des schémas spécifiques et d'appliquer le balisage adéquat. Elle garantit également une flexibilité, permettant à l'algorithme de s'ajuster face à de nouvelles normes ou exigences de balisage. De plus, cela contribue à la réduction significative des erreurs de balisage ou des omissions. La mise en place d'une telle automatisation nécessiterait une collaboration étroite avec des experts, ainsi que la collecte d'un ensemble de données représentatif pour l'entraînement, afin de s'assurer que l'algorithme respecte les normes de l'industrie et répond aux besoins spécifiques du projet.

Conclusion

L'ère numérique a profondément modifié notre interaction avec le patrimoine historique, et le projet AGODA en est une parfaite illustration. Si la numérisation des débats parlementaires est déjà un exploit en soi, c'est véritablement dans les méthodologies innovantes adoptées pour rendre ces archives accessibles et exploitables que réside la prouesse. Le projet AGODA a mis un accent particulier sur l'automatisation des processus. Face à plus de 10 000 images numérisées à traiter, l'automatisation n'était pas un luxe, mais une nécessité. La mise en place des commentaires lors de l'océrisation, ainsi que l'automatisation du processus d'encodage, ont montré leur capacité à traiter une grande quantité de données avec efficacité et précision, minimisant les erreurs humaines et accélérant la mise à disposition des archives. La transformation des images numérisées en textes éditables grâce à l'encodage automatique a été une étape clé. Ce processus, transformant les données en formats exploitables comme le XML est fondamental. Il structure l'information, la rendant interrogeable, et ouvre la voie à une multitude d'analyses. L'ajout de commentaires automatisés lors de l'océrisation peut être envisagé et facilitera véritablement le traitement automatique du balisage.

Ainsi, le projet AGODA, ancré dans le domaine des humanités numériques, met en avant le potentiel de la technologie pour transformer et enrichir la recherche en humanités. La mise en œuvre de l'encodage, malgré ses défis inhérents, montre que l'automatisation, lorsqu'elle est bien exécutée, peut grandement simplifier des tâches qui étaient autrefois manuelles et fastidieuses. La qualité du fichier OCR et la richesse des annotations sont des éléments clés pour faciliter ce processus.

Cependant, la technologie a ses limites. Bien que les fichiers XML générés soient un grand pas en avant pour le projet, ils nécessiteront des ajustements pour répondre aux normes élevées requises pour la publication académique ou institutionnelle. De plus, le script actuel, malgré son efficacité, peut être optimisé pour une meilleure lisibilité, maintenance, et réutilisabilité.

L'expérience d'AGODA souligne l'importance d'une approche critique en matière d'humanités numériques. Alors que les outils numériques offrent d'immenses possibilités, leur utilisation nécessite une réflexion et une compréhension approfondies. En fin de compte, avec des ajustements appropriés, le projet AGODA est bien positionné pour servir de modèle pour d'autres initiatives dans le champ des humanités numériques. En partageant les connaissances et les meilleures pratiques, la communauté peut progresser de manière plus intégrée et efficace.

Bibliographie

- André, J. (2003). Numérisation et codage des caractères de livres anciens. *Document numérique*, 7(3-4), 127-142. <https://doi.org/10.3166/dn.7.3-4.127-142>
- Bachimont, B. (2007). *Chapitre 1. Nouvelles tendances applicatives : De l'indexation à l'éditorialisation - PDF Free Download*. <https://docplayer.fr/127830894-Chapitre-1-nouvelles-tendances-applicatives-de-l-indexation-a-l-editorialisation.html>
- Barbier, J., & Mandret-Degeilh, A. (2018). Les archives numériques et numérisées. In *Le travail sur archives* (p. 195-222). Armand Colin. <https://www.cairn.info/le-travail-sur-archives--9782200621056-p-195.htm>
- BLUM, C. (2008). *Le Journal officiel dans Gallica (1869-1946) | Le blog de Gallica*. <https://gallica.bnf.fr/blog/19012016/le-journal-officiel-dans-gallica-1869-1946?mode=desktop>
- BNF. (1889, novembre 26). *Journal officiel de la République française. Débats parlementaires. Chambre des députés : Compte rendu in-extenso*. Gallica. <https://gallica.bnf.fr/ark:/12148/bpt6k64948012>
- Bourgeois, N., Pellet, A., & Puren, M. (2022). Using Topic Generation Model to explore the French Parliamentary Debates during the early Third Republic (1881-1899). In M. L. Mela, F. Norén, & E. Hyvönen (Éds.), *DiPaDA 2022 Digital Parliamentary Data in Action 2022. Proceedings of the Digital Parliamentary Data in Action (DiPaDA 2022) Workshop co-located with 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022)* (Vol. 3133, p. 35-51). <https://hal.science/hal-03526254>
- Bouzidi, L., & Boulesnane, S. (2017). Les humanités numériques. L'évolution des usages et des pratiques. *Les Cahiers du numérique*, 13(3-4), 19-38.
- Buard, P.-Y. (2015). *Modélisation des sources anciennes et édition numérique* [Phdthesis, Université de Caen]. <https://hal.science/tel-01279385>
- Bureau, B. (2019). Les enjeux nouveaux qu'ouvrent les Humanités Numériques en lien avec les pratiques éditoriales traditionnelles et leur questionnement. *Bulletin de l'Association Guillaume Budé*, 1(1), 132-149. <https://doi.org/10.3406/bude.2019.7279>
- Burnard, L. (2015a). La TEI et le XML. In *Qu'est-ce que la Text Encoding Initiative ?* OpenEdition Press. <https://doi.org/10.4000/books.oep.1298>

- Burnard, L. (2015b). Qu'est-ce que la Text Encoding Initiative ? In M. Burghart (Trad.), *Qu'est-ce que la Text Encoding Initiative ?* OpenEdition Press. <https://doi.org/10.4000/books.oep.1237>
- Carlin, M., & Laborderie, A. (2021). Le BnF DataLab, un service aux chercheurs en humanités numériques. *Humanités numériques*, 4, Article 4. <https://doi.org/10.4000/revuehn.2684>
- Cincinnatus. (2020). Vous avez dit : « humanités numériques » ? *Humanisme*, 326(1), 4-8. <https://doi.org/10.3917/huma.326.0004>
- Coniez, H. (2010). L'Invention du compte rendu intégral des débats en France (1789-1848). *Parlement[s], Revue d'histoire politique*, 14(2), 146-158. <https://doi.org/10.3917/parl.014.0146>
- Dacos, M., & Mounier, P. (2010a). *Conclusion / Les cinq piliers de l'édition électronique. À nouveaux enjeux, nouveaux métiers* (p. 108-113). La Découverte. <https://www.cairn.info/l-edition-electronique--9782707157294-p-108.htm>
- Dacos, M., & Mounier, P. (2010b). *III. L'édition au défi du numérique* (p. 49-65). La Découverte. <https://www.cairn.info/l-edition-electronique--9782707157294-p-49.htm>
- Delente, É., & Renault, R. (2023). Traitement automatique des textes versifiés : Problématiques et pratiques. *Langages*, 199, 1-11.
- Diwersy, S., Frontini, F., & Luxardo, G. (2018). *The Parliamentary Debates as a Resource for the Textometric Study of the French Political Discourse*. Proceedings of the ParlaCLARIN@LREC2018 workshop. <https://hal.science/hal-01832649>
- Diwersy, S., & Luxardo, G. (2020). Querying a large annotated corpus of parliamentary debates. *Proceedings of the Second ParlaCLARIN Workshop*, 75-79. <https://aclanthology.org/2020.parlaclarin-1.13>
- Gardey, D. (2010). Scriptes de la démocratie : Les sténographes et rédacteurs des débats (1848–2005). *Sociologie du travail*, 52(2), Article 2. <https://doi.org/10.4000/sdt.13695>
- Gaudillère, B. (2008). La publicité des débats parlementaires (1852-1870). *Parlement[s], Revue d'histoire politique*, HS 4(3), 27-49. <https://doi.org/10.3917/parl.hs04.0027>
- Hansard—UK Parliament. (s. d.). Consulté 26 août 2023, à l'adresse [https://hansard.parliament.uk/Journal officiel de la République française. Lois et décrets. \(1889, novembre 26\). Gallica.](https://hansard.parliament.uk/Journal%20officiel%20de%20la%20République%20française.Lois%20et%20décrets.(1889,%20novembre%2026).Gallica) <https://gallica.bnf.fr/ark:/12148/bpt6k6452360g>
- Lavoinnie, Y. (2022). Publicité des débats et espace public. *Études de communication. langages, information, médiations*, Article 22. <https://doi.org/10.4000/edc.2350>
- Lebreton, F. (2023). *Vers l'ouverture et l'exploration des débats parlementaires : Étude d'une méthodologie de structuration et d'enrichissement automatique des données*. <https://github.com/FannyLbr/Memoire-AGODA-TNAH2022> (Édition originale 2022)

- Lebreton, F., Puren, M., & Vernus, P. (s. d.). *AGODA : Schéma TEI pour les débats parlementaires français de la Chambre des députés* [Text]. Consulté 11 août 2023, à l'adresse https://agoda-project.github.io/agoda_odd.html
- Lemay, Y., & Klein, A. (2012). La diffusion des archives ou les 12 travaux des archivistes à l'ère du numérique. *Les Cahiers du numérique*, 8(3), 15-48. <https://doi.org/10.3166/LCN.8.3.15-48>
- Likforman-Sulem, L. (2003). Apport du traitement des images à la numérisation des documents manuscrits anciens. *Document numérique*, 7(3-4), 13-26. <https://doi.org/10.3166/dn.7.3-4.13-26>
- Lipsyc, C., & Ihadjadene, M. (2013). Architecture de l'information et éditorialisation. *Études de communication. langages, information, médiations*, 41, Article 41. <https://doi.org/10.4000/edc.5406>
- Longhi, J. (2017). Humanités, numérique : Des corpus au sens, du sens aux corpus. *Questions de communication*, 31(1), 7-17. <https://doi.org/10.4000/questionsdecommunication.11039>
- Malais, N. (2014). Prescrire à Babel : Prescription et numérisation du patrimoine. *Communication & langages*, 179(1), 91-104. <https://doi.org/10.3917/comla.179.0091>
- Malrieu, D. (2005). *Quel balisage du roman contemporain?*
- Mechri, H. (2017, décembre 12). *Balisage des guides de bonnes pratiques : Solutions, standards et outils disponibles.*
- Meunier, J.-G. (2017). Humanités numériques et modélisation scientifique. *Questions de communication*, 31(1), 19-48. <https://doi.org/10.4000/questionsdecommunication.11040>
- Morel, B. (2018). Ce que conte le compte rendu : L'institution d'un ordre parlementaire idéalisé. *Droit et société*, 98(1), 179-199. <https://doi.org/10.3917/drs.098.0179>
- Pellet, A., Lebreton, F., Bourgeois, N., Vernus, P., & Puren, M. (2022, mai). Analysis of the French Parliamentary Debates of the Third Republic with Topic Modelling and Word Embedding. Methodological Challenges and First Results. *Journées MASHS (Modèles et Apprentissages en Sciences Humaines et Sociales)*. <https://hal.science/hal-03682991>
- Pellet, A., & Puren, M. (2022, avril). Le projet AGODA. Océrisation des débats parlementaires français de la Troisième République : Problèmes, défis et perspectives. *Séminaire OMNSH-Epitech : le numérique au service des sciences humaines et sociales*. <https://hal.science/hal-03651146>
- Puren, M. (2022). *AGODA* [Jupyter Notebook]. <https://github.com/mpuren/agoda> (Édition originale 2021)

- Puren, M., Pellet, A., Bourgeois, N., Vernus, P., & Lebreton, F. (2022, juin). Between History and Natural Language Processing: Study, Enrichment and Online Publication of French Parliamentary Debates of the Early Third Republic (1881-1899). *ParlaCLARIN III at LREC2022 - Workshop on Creating, Enriching and Using Parliamentary Corpora*. <https://hal.science/hal-03623351>
- Puren, M., & Vernus, P. (2021, octobre). AGODA : Analyse sémantique et Graphes relationnels pour l'Ouverture et l'étude des Débats à l'Assemblée nationale. *Inauguration du BnF DataLab*. <https://hal.science/hal-03382765>
- Puren, M., Vernus, P., Pellet, A., Bourgeois, N., & Lebreton, F. (2022a, mai 19). *Le projet AGODA. Annoter et publier les débats parlementaires français de la fin du XIXe siècle: Défis et solutions*. Colloque Humanistica 2022. <https://hal.science/hal-03674919>
- Puren, M., Vernus, P., Pellet, A., Bourgeois, N., & Lebreton, F. (2022b, juin). Extracting and providing online access to annotated and semantically enriched historical data. The AGODA project. *DH Benelux 2022*. <https://hal.science/hal-03683018>
- Renouard, C. P. (2021). *Marie Puren (Epitech) Pierre Vernus (LARHRA)*.
- Roland, L. (2014). *STRUCTUREZ VOS DONNEES AVEC XML* (Open Classrooms). <https://unr-rascholarvox-com.bibelec.univ-lyon2.fr/catalog/search/searchterm/xml?searchtype=all>
- Salah, A. B. (2014). *Maîtrise de la qualité des transcriptions numériques dans les projets de numérisation de masse* [Phdthesis, Université de Rouen]. <https://bnf.hal.science/tel-01164698>
- Sandras, A. (2020). Les images des imprimés de la BNF : Comment guider les usagers au sein d'un « eldorado numérique » ? *Sociétés & Représentations*, 50(2), 77-93. <https://doi.org/10.3917/sr.050.0077>
- Saudrais, H. (2015). Aux sources de la loi, les archives parlementaires (XIXe-XXe siècles). *Revue française de droit constitutionnel*, 101(1), 165-175. <https://doi.org/10.3917/rfdc.101.0165>
- van Hooland, S., Gillet, F., Hengchen, S., & De Wilde, M. (2016a). Chapitre 1. Trouver l'information. In *Introduction aux humanités numériques : Méthodes et pratiques* (p. 15-40). De Boeck Supérieur. <https://www.cairn.info/introduction-aux-humanites-numeriques-methodes--9782807302150-p-15.htm>
- van Hooland, S., Gillet, F., Hengchen, S., & De Wilde, M. (2016b). Chapitre 3. Numériser les sources. In *Introduction aux humanités numériques : Méthodes et pratiques* (p. 81-123). De Boeck Supérieur. <https://www.cairn.info/introduction-aux-humanites-numeriques-methodes--9782807302150-p-81.htm>

Verlaet, L. (2010). Application du Web sémantique : Vers l'avènement du balisage sémantique et des modélisations des connaissances évolutives ? *Journal of Media Research - Revista de Studii Media*, 12-24.

Annexes

- Les documents techniques tels que les scripts, le guide d'encodage, et les fichiers xml générés seront joint au mail
- Les extraits de code, captures d'écran, etc sont directement intégrés dans le mémoire

Merci de votre lecture !