Vers l'ouverture et l'exploration des débats parlementaires : étude d'une méthodologie de structuration et d'enrichissement automatique des données. L'exemple des débats à la Chambre des députés durant la Ve législature de la IIIe République (1889-1893)

Fanny Lebreton

ÉCOLE NATIONALE DES CHARTES
UNIVERSITÉ PARIS, SCIENCES & LETTRES

Fanny Lebreton
Licenciée ès lettres et arts

Vers l'ouverture et l'exploration des débats parlementaires : étude d'une méthodologie de structuration et d'enrichissement automatique des données

L'exemple des débats à la Chambre des députés durant la V^{e} législature de la III^{e} République (1889-1893)

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

2022

Résumé

Ce mémoire a été réalisé à l'issue d'un stage de quatre mois effectué au sein de l'équipe du projet AGODA, dans le cadre du Master « Technologies numériques appliquées à l'histoire » de l'École nationale des chartes. Il porte sur l'étude d'une méthodologie de structuration et d'enrichissement automatique des données, appliquée aux comptes rendus *in extenso* des débats parlementaires de la Chambre des députés, de la V^{e} législature de la III^{e} République. Le mémoire apporte une analyse sur les stratégies mises en place et les résultats obtenus durant le stage, au regard des enjeux du projet AGODA.

Mots-clés : AGODA ; débats parlementaires ; compte rendu *in extenso* ; corpus ; chaîne de traitement ; encodage ; balisage automatique ; programmation ; JSON ; XML-TEI ; ODD ; Roma ; Python.

Informations bibliographiques : Fanny Lebreton, *Vers l'ouverture et l'exploration des débats parlementaires : étude d'une méthodologie de structuration et d'enrichissement automatique des données. L'exemple des débats à la Chambre des députés durant la V^{e} législature de la III^{e} République (1889-1893)*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Ségolène Albouy, École nationale des chartes, 2022.

Remerciements

Je tiens à remercier toutes les personnes qui m'ont apporté leur savoir, leur soutien et leur encouragement tout au long de cette dernière année d'étude.

Je remercie tout d'abord mes tuteurs de stage, Madame Marie Puren et Monsieur Pierre Vernus pour leur encadrement, leur disponibilité, et leur bienveillance durant ce stage.

Je souhaite remercier également ma directrice de mémoire, Madame Ségolène Albouy, pour son aide précieuse durant la réalisation de mes missions de stage.

Je remercie aussi les membres du laboratoire Méthodes Numériques pour les Sciences de l'Humain et de la Société (MNSHS) pour leur accueil et leur gentillesse, et notamment Aurélien Pellet et Nicolas Bourgeois qui ont su se rendre disponibles et m'apporter un appui bénéfique lorsque je bloquais sur mes lignes de code.

Mes remerciements vont également à l'équipe pédagogique de l'École nationale des chartes pour son enseignement durant ces deux années de master.

Enfin, je remercie mon entourage, tout particulièrement mes parents qui m'ont soutenue et motivée tout au long de mes années d'études, mes amies Cécile et Lien, pour leur épaulement quotidien au sein de la promotion TNAH, et Maxime, pour son soutien sans faille, sans qui je n'aurais accompli tout ce que j'ai fait.

Bibliographie

Sources primaires

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.

Journal Officiel de La République Française - 70 Années Disponibles, Gallica, URL : https://gallica.bnf.fr/ark:/12148/cb34378481r/date.r=Journal+officiel+de+la+republique+francaise+Lois+et+decrets.langFR (visité le 14/08/2022).

Histoire des débats

Blum (Catherine), *Le Journal Officiel Dans Gallica (1869-1946)*, Le blog de Gallica, 19 janv. 2016, URL : https://gallica.bnf.fr/blog/19012016/le-journal-officiel-dans-gallica-1869-1946?mode=desktop (visité le 11/08/2022).

Coniez (Hugo), « L'Invention du compte rendu intégral des débats en France (1789-1848) », *Parlement[s], Revue d'histoire politique* (, 2010), p. 146-158, URL : https://www.cairn.info/revue-parlements1-2010-2-page-146.htm (visité le 12/08/2022).

Gardey (Delphine), « Scriptes de la démocratie : les sténographes et rédacteurs des débats (1848–2005) », *Sociologie du travail*–2 (1er juin 2010), p. 195-211, URL : https://journals.openedition.org/sdt/13695 (visité le 12/08/2022).

Gaudillère (Bernard), « La publicité des débats parlementaires (1852-1870) », *Parlement[s], Revue d'histoire politique* (, 2008), p. 27-49, URL : https://www.cairn.info/revue-parlements1-2008-3-page-27.htm (visité le 12/08/2022).

Journal officiel de la République française, dans *Wikipédia*, 2022, URL : https://fr.wikipedia.org/w/index.php?title=Journal_officiel_de_la_R%C3%A9publique_fran%C3%A7aise&oldid=195403573 (visité le 13/08/2022).

Lavoinne (Yves), « Publicité des débats et espace public », *Études de communication. langages, information, médiations*–22 (1er déc. 1999), p. 115-132, URL : https://journals.openedition.org/edc/2350 (visité le 12/04/2022).

Le Torrec (Virginie), « Aux frontières de la publicité parlementaire : les assemblées et leur visibilité mediatisée », *Réseaux* (, 2005), p. 181-208, URL : https://www.cairn.info/revue-reseaux1-2005-1-2-page-181.htm (visité le 12/08/2022).

Martin (H.-Marie), « Variétés. Nouveau manuel de sténographie ou Art de suivre la parole en écrivant », *Le Constitutionnel : journal du commerce, politique et littéraire* (, 24 janv. 1867), URL : https://gallica.bnf.fr/ark:/12148/bpt6k674517j (visité le 12/08/2022).

Mayeur (Jean Marie), *La Vie Politique Sous La Troisième République : 1870-1940*, Paris, 1984 (Points. Histoire).

Morel (Benjamin), « Ce que conte le compte rendu : l'institution d'un ordre parlementaire idéalisé », *Droit et société* (, 2018), p. 179-199, URL : https://www.cairn.info/revue-droit-et-societe-2018-1-page-179.htm (visité le 31/08/2022).

Poudra (Jules) et Pierre (Eugène), *Traité pratique de droit parlementaire*, 1878, URL : https://gallica.bnf.fr/ark:/12148/bpt6k58651348 (visité le 12/08/2022).

Prévost (Hippolyte), *Nouveau manuel de sténographie ou Art de suivre la parole en écrivant*, 4e éd. revue, augmentée et accompagnée de planches, 1834, URL : https://gallica.bnf.fr/ark:/12148/bpt6k2991973 (visité le 12/08/2022).

— « Variétés. Organisation de la sténographie officielle de l'Assemblée nationale. » *Le Constitutionnel : journal du commerce, politique et littéraire* (, 19 juin 1848), URL : https://gallica.bnf.fr/ark:/12148/bpt6k668244c (visité le 13/08/2022).

Rozenberg (Olivier) et Baudot (Pierre-Yves), « Entretien avec Claude Azéma, directeur du service du compte rendu intégral à l'Assemblée nationale », *Parlement[s], Revue d'histoire politique* (, 2010), p. 133-145, URL : https://www.cairn.info/revue-parlements1-2010-2-page-133.htm (visité le 12/08/2022).

Saudrais (Hélène), « Aux sources de la loi, les archives parlementaires (XIXe-XXe siècles) », *Revue française de droit constitutionnel* (, 2015), p. 165-175, URL : https://www.cairn.info/revue-francaise-de-droit-constitutionnel-2015-1-page-165.htm (visité le 12/08/2022).

Travaux mobilisant les débats : AGODA

Bourgeois (Nicolas), Pellet (Aurélien) et Puren (Marie), « Using Topic Generation Model to Explore the French Parliamentary Debates during the Early Third Republic (1881-1899) », dans *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)*, dir. Matti La Mela, Fredrik Norén et Eero Hyvönen, 2022 (CEUR Workshop Proceedings), p. 35-51, URL : https://hal.archives-ouvertes.fr/hal-03526254 (visité le 09/08/2022).

Lebreton (Fanny), Puren (Marie) et Vernus (Pierre), *AGODA : Schéma TEI Pour Les Débats Parlementaires Français de La Chambre Des Députés*, juill. 2022, url : https://agoda-project.github.io/agoda_odd.html (visité le 04/09/2022).

Pellet (Aurélien), Lebreton (Fanny), Bourgeois (Nicolas), Vernus (Pierre) et Puren (Marie), « Analysis of the French Parliamentary Debates of the Third Republic with Topic Modelling and Word Embedding. Methodological Challenges and First Results », dans 2022, url : https://hal.archives-ouvertes.fr/hal-03682991 (visité le 09/08/2022).

Pellet (Aurélien) et Puren (Marie), « Le projet AGODA. Océrisation des débats parlementaires français de la Troisième République : problèmes, défis et perspectives », dans 2022, url : https://hal.archives-ouvertes.fr/hal-03651146 (visité le 09/08/2022).

Puren (Marie), *AGODA*, avec la coll. de Pierre Vernus, Aurélien Pellet et Fanny Lebreton, url : https://github.com/mpuren/agoda (visité le 14/08/2022).

Puren (Marie), Bourgeois (Nicolas), Pellet (Aurélien), Vernus (Pierre) et Lebreton (Fanny), « Between History and Natural Language Processing : Study, Enrichment and Online Publication of French Parliamentary Debates of the Early Third Republic (1881-1899) », dans *ParlaCLARIN III at LREC2022 - Workshop on Creating, Enriching and Using Parliamentary Corpora*, Marseille, France, 2022, url : https://hal.archives-ouvertes.fr/hal-03623351 (visité le 09/08/2022).

Puren (Marie) et Vernus (Pierre), « AGODA : Analyse Sémantique et Graphes Relationnels Pour l'Ouverture et l'étude Des Débats à l'Assemblée Nationale », dans *Inauguration Du BnF DataLab*, Paris, France, 2021, url : https://hal.archives-ouvertes.fr/hal-03382765 (visité le 06/04/2022).

Puren (Marie), Vernus (Pierre), Pellet (Aurélien), Bourgeois (Nicolas) et Lebreton (Fanny), « Extracting and Providing Online Access to Annotated and Semantically Enriched Historical Data. The AGODA Project », dans *DH Benelux 2022*, Luxembourg, Luxembourg, 2022, url : https://hal.archives-ouvertes.fr/hal-03683018 (visité le 09/08/2022).

— « Le Projet AGODA. Annoter et Publier Les Débats Parlementaires Français de La Fin Du XIXe Siècle : Défis et Solutions », dans *Colloque Humanistica 2022*, Montréal, Canada, 2022, url : https://hal.archives-ouvertes.fr/hal-03674919 (visité le 09/08/2022).

Travaux mobilisant les débats : autres projets

Diwersy (Sascha), Frontini (Francesca) et Luxardo (Giancarlo), « The Parliamentary Debates as a Resource for the Textometric Study of the French Political Discourse », dans *Proceedings of the ParlaCLARIN@LREC2018 Workshop*, Miyazaki,

Japan, 2018, URL : https://hal.archives-ouvertes.fr/hal-01832649 (visité le 10/03/2022).

DIWERSY (Sascha) et LUXARDO (Giancarlo), « Querying a Large Annotated Corpus of Parliamentary Debates », dans *Proceedings of the Second ParlaCLARIN Workshop*, Marseille, France, 2020, p. 75-79, URL : https://aclanthology.org/2020.parlaclarin-1.13 (visité le 10/03/2022).

Hansard - UK Parliament, URL : https://hansard.parliament.uk// (visité le 04/09/2022).

LANDOWSKI (Éric), « Le débat parlementaire et l'écriture de la loi », *Revue française de science politique* (, 1977), p. 428-441, URL : https://www.persee.fr/doc/rfsp_0035-2950_1977_num_27_3_418267 (visité le 21/08/2022).

LEMERCIER (Claire), « Un catholique libéral dans le débat parlementaire sur le travail des enfants dans l'industrie (1840) », *Parlement[s], Revue d'histoire politique* (, 2021), p. 195-206, URL : https://www.cairn.info/revue-parlements-2021-1-page-195.htm (visité le 13/08/2022).

MARNOT (Bruno), *Les ingénieurs au Parlement sous la IIIe République*, 2000, URL : https://gallica.bnf.fr/ark:/12148/bpt6k3373008g (visité le 13/08/2022).

OUELLET (Jérôme) et ROUSSEL-BEAULIEU (Frédéric), « Les débats parlementaires au service de l'histoire politique », *Bulletin d'histoire politique* (, 2003), p. 23-40, URL : https://www.erudit.org/fr/revues/bhp/2003-v11-n3-bhp04658/1060736ar/resume/ (visité le 12/08/2022).

RHEAULT (Ludovic), BEELEN (Kaspar), COCHRANE (Christopher) et HIRST (Graeme), « Measuring Emotion in Parliamentary Debates with Automated Textual Analysis », *PLOS ONE* (, 22 déc. 2016), URL : https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0168843 (visité le 13/08/2022).

TRUAN (Naomi), *"Who Are You Talking About ?". The Pragmatics of Third-Person Referring Expressions : A Contrastive Corpus-Based Study of British, German, and French Parliamentary Debates*, These de doctorat, Sorbonne université, 2019, URL : http://www.theses.fr/2019SORUL014 (visité le 15/03/2022).

— « Talking about, for, and to the People : Populism and Representation in Parliamentary Debates on Europe », *Zeitschrift für Anglistik und Amerikanistik* (, 25 sept. 2019), p. 307-337, URL : https://www.degruyter.com/document/doi/10.1515/zaa-2019-0025/html (visité le 13/08/2022).

TRUAN (Naomi) et ROMARY (Laurent), « Building, Encoding, and Annotating a Corpus of Parliamentary Debates in XML-TEI : A Cross-Linguistic Account », *Journal of the Text Encoding Initiative* (, 2021), URL : https://halshs.archives-ouvertes.fr/halshs-03097333 (visité le 14/08/2022).

Humanités numériques et édition électronique

Bachimont (Bruno), « Chapitre 1. Nouvelles tendances applicatives : de l'indexation à l éditorialisation », dans Patrick Gros, *L'indexation multimédia : description et recherche automatiques*, Paris, 2007, url : https://docplayer.fr/127830894-Chapitre-1-nouvelles-tendances-applicatives-de-l-indexation-a-l-editorialisation.html.

Clavert (Frédéric), « Vers de nouveaux modes de lecture des sources », dans Olivier Le Deuff, *Le temps des humanités digitales : la mutation des sciences humaines et sociales*, fyp, 2017, p. 33-47.

Dacos (Marin) et Mounier (Pierre), *L'édition électronique*, Paris, 2010 (Repères, 549).

— *Humanités Numériques : État Des Lieux et Positionnement de La Recherche Française Dans Le Contexte International*, Research Report, Institut français, 2015, p. 9782354761097, url : https://hal.archives-ouvertes.fr/hal-01228945 (visité le 11/08/2022).

Longhi (Julien), « Humanités, numérique : des corpus au sens, du sens aux corpus », *Questions de communication*–31 (1er sept. 2017), url : https://journals.openedition.org/questionsdecommunication/11039 (visité le 11/09/2022).

Moretti (Franco), *Distant Reading*, London ; New York, 2013.

Mounier (Pierre), *Les humanités numériques : Une histoire critique*, Paris, 2018 (Interventions), url : http://books.openedition.org/editionsmsh/12006 (visité le 11/08/2022).

Poupeau (Gautier), « L'édition électronique change tout et rien. Dépasser les promesses de l'édition électronique », *Le médiéviste et l'ordinateur* (, 18 avr. 2004), url : https://archivesic.ccsd.cnrs.fr/sic_00137222 (visité le 10/08/2022).

Rosati (Marcello Vitali) et Sinatra (Michael E.), « Introduction », dans *Pratiques de l'édition numérique*, 2014, url : http://parcoursnumeriques-pum.ca/1-pratiques/%E2%80%8Bhttps://www.parcoursnumeriques-pum.ca/1-pratiques/introduction.html (visité le 11/09/2022).

Sinatra (Michaël E.) et Vitali-Rosati (Marcello), « Chapitre 3. Histoire Des Humanités Numériques », dans *Pratiques de l'édition Numérique*, Montréal, 2014 (Parcours Numérique), p. 49-60, url : http://books.openedition.org/pum/317 (visité le 10/08/2022).

Numérisation et océrisation

Andro (Mathieu), « Actualité de La Numérisation », *Bulletin des bibliothèques de France* (, 2011), p. 27-29, url : https://hal.archives-ouvertes.fr/hal-01094553 (visité le 11/08/2022).

Belaïd (Abdel), Pierron (Laurent), Najman (Laurent) et Reyren (Dominique), *La numérisation de documents : principes et évaluation des performances*, 2000, 35 p, url : https://hal.inria.fr/inria-00099148 (visité le 11/08/2022).

Ben Salah (Ahmed), *Maîtrise de La Qualité Des Transcriptions Numériques Dans Les Projets de Numérisation de Masse*, Theses, Université de Rouen, 2014, url : https://hal-bnf.archives-ouvertes.fr/tel-01164698 (visité le 11/08/2022).

Droit, économie, politique et presse dans Gallica, BnF - Site institutionnel, url : https://www.bnf.fr/fr/droit-economie-politique-et-presse-dans-gallica (visité le 11/08/2022).

Gallica, la Bibliothèque numérique de la BnF et de ses partenaires, BnF - Site institutionnel, url : https://www.bnf.fr/fr/gallica-la-bibliotheque-numerique-de-la-bnf-et-de-ses-partenaires (visité le 11/08/2022).

Encodage des données

Albouy (Ségolène), *Décrire Les Documents Patrimoniaux - Cours*, 25 oct. 2021, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/03-XML_TEI/2021-10-25-TEI.pdf (visité le 14/08/2022).

— *Introduction à XML - Cours*, 13 oct. 2021, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/3f4cb64280c009553e0db6c820ae228893699695/01-Introduction_XML/2021-10-13-Introduction_XML.pdf (visité le 09/09/2022).

— *Les TEI Guidelines - Cours*, 22 nov. 2021, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/e7c2aabedd885b98fbf11faacea32ff707eb01af/05-Les_TEI_Guidelines/2021-11-22-Guidelines.pdf (visité le 01/09/2022).

— *Structure Du Document TEI - Cours*, 15 nov. 2021, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/04-Structure_document_TEI/2021-11-15-Structure_TEI.pdf (visité le 14/08/2022).

— *Les Schémas XML - Cours*, 10 janv. 2022, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/10-ODD/2022-01-10-Introduction_ODD.pdf (visité le 14/08/2022).

— *Les Schémas XML 2 - Cours*, 17 janv. 2022, url : https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/11-Personnalisation_ODD/2022-01-17-Personnalisation_ODD.pdf (visité le 14/08/2022).

Burnard (Lou), *Qu'est-ce que la Text Encoding Initiative ?*, Marseille, 2015 (Encyclopédie numérique), url : http://books.openedition.org/oep/1237 (visité le 10/08/2022).

Calderan (Lisette), Hidoine (Bernard) et Millet (Jacques), *Métadonnées : Mutations et Perspectives : Séminaire INRIA, 29 Septembre-3 Octobre 2008, Dijon*, Paris, 2008 (Sciences et Techniques de l'information).

Clarin (Éric), *Parla-CLARIN*, GitHub, URL : https://github.com/clarin-eric/parla-clarin (visité le 14/08/2022).

— *ParlaMint*, GitHub, URL : https://github.com/clarin-eric/ParlaMint (visité le 14/08/2022).

Erjavec (Tomaž) et Pančur (Andrej), *Parla-CLARIN : TEI Guidelines for Corpora of Parliamentary Proceedings*, 19 sept. 2019, URL : https://zenodo.org/record/3446164 (visité le 14/08/2022).

Erjavec Tomaž, Ogrodniczuk Maciej, Osenova Petya, Ljubešić Nikola, Simov Kiril, Pančur Andrej, Rudolf Michał, Kopp Matyáš, Barkarson Starkaður, Steingrímsson Steinþór, *et al.*, « The ParlaMint Corpora of Parliamentary Proceedings », *Language Resources and Evaluation* (, 2 févr. 2022), URL : https://doi.org/10.1007/s10579-021-09574-0 (visité le 14/08/2022).

Pančur (Andrej) et Erjavec (Tomaž), *Parla-CLARINA TEI Schema for Corpora of Parliamentary Proceedings*, 3 mai 2022, URL : https://clarin-eric.github.io/parla-clarin/ (visité le 14/08/2022).

— *The Structure and Encoding of ParlaMint Corpora*, 27 juill. 2022, URL : https://clarin-eric.github.io/ParlaMint/ (visité le 30/08/2022).

Automatisation de l'encodage

Clérice (Thibault), *Introduction à Python et Au Développement Web Avec Python Pour Les Sciences Humaines*, 1er août 2022, URL : https://github.com/PonteIneptique/cours-python (visité le 14/08/2022).

Datetime — Basic Date and Time Types — Python 3.10.6 Documentation, URL : https://docs.python.org/3/library/datetime.html (visité le 14/08/2022).

Json — JSON Encoder and Decoder — Python 3.10.6 Documentation, URL : https://docs.python.org/3/library/json.html (visité le 14/08/2022).

Le Fourner (Victoria), *Étude de La Structuration Automatique et de l'éditorialisation d'un Corpus Hétérogène*, Technical Report, Ecole nationale des Chartes, 2019, URL : https://hal.archives-ouvertes.fr/hal-03577063 (visité le 14/08/2022).

Lxml - Processing XML and HTML with Python, URL : https://lxml.de/ (visité le 14/08/2022).

Os — Diverses Interfaces Pour Le Système d'exploitation — Documentation Python 3.10.6, URL : https://docs.python.org/fr/3/library/os.html (visité le 14/08/2022).

Re — Regular Expression Operations — Python 3.10.6 Documentation, URL : https://docs.python.org/3/library/re.html (visité le 14/08/2022).

Time Us · GitLab, GitLab, URL : https://gitlab.inria.fr/almanach/time-us (visité le 14/08/2022).

Tschirhart (Daniel), « Quelques expressions régulières simples (1) » (), p. 16.

Logiciels services et plateformes

Dib (Firas), *Regex101 : Build, Test, and Debug Regex*, regex101, URL : `https://regex101.com/` (visité le 14/08/2022).

GitHub, GitHub, URL : `https://github.com` (visité le 14/08/2022).

Oxygen XML Editor, URL : `https://www.oxygenxml.com/` (visité le 14/08/2022).

PyCharm, JetBrains, URL : `https://www.jetbrains.com/fr-fr/pycharm/` (visité le 14/08/2022).

Roma : Generating Customizations for the TEI, URL : `https://roma.tei-c.org/` (visité le 14/08/2022).

Sublime Text, URL : `https://www.sublimetext.com/` (visité le 14/08/2022).

TEI Publisher, URL : `https://teipublisher.com/index.html` (visité le 14/08/2022).

TEI : Text Encoding Initiative, URL : `https://tei-c.org/` (visité le 14/08/2022).

Introduction

« La séance est ouverte… » : ces premiers mots, prononcés en ouverture des séances parlementaires, marquent le début des débats à l'Assemblée nationale. Délibérer et voter sur des sujets de société (social, économique, culturel, religieux, etc.), telles sont les missions des représentants de la Nation lors de ces séances. Celles-ci jouent un rôle politique considérable et constituent le cœur du travail parlementaire.

Véritable temps de parole où l'Histoire se fait, les débats parlementaires intéressent l'opinion publique. Au cours du XIX^{e} siècle, les séances sont devenues publiques, et les propos tenus ont fait l'objet de publications diverses et variées. Dès lors, les débats parlementaires sont devenus accessibles au plus grand nombre. Lors de la III^{e} République, les débats parlementaires ont été publiés de façon officielle sous la forme de comptes rendus *in extenso* dans le *Journal Officiel de la République française*. Cette publication, toujours diffusée aujourd'hui, donne à lire l'intégralité des débats parlementaires, relayant aux citoyens à la fois le naturel des délibérations et le résultat de ces dernières. Les comptes rendus ont ainsi pour rôle d'informer les citoyens.

Ils permettent également de garder une trace des questionnements politiques d'une époque donnée. Comme l'affirmait H.-Marie Martin dans le numéro du 24 janvier 1867 du journal *Le Constitutionnel* : « Le compte-rendu officiel profite donc au présent ; il profitera aussi à l'histoire [1] ». Les comptes rendus *in extenso* du passé constituent ainsi une source riche pour l'historien d'aujourd'hui. Ils permettent de comprendre les étapes majeures d'élaboration du cadre législatif des différents champs d'activités. Ces « plus importants souvenirs [2] » sont une source d'étude incontournable pour l'histoire politique et les autres champs historiques, mais aussi une source intéressante pour la science politique, la sociologie et la linguistique. Cependant, bien que les comptes rendus des débats parlementaires soient utiles pour la recherche et riches d'informations, ils restent encore sous-étudiés par les chercheurs, et très peu consultés par le grand public. Hugo Coniez soutient en effet à ce sujet : « Les comptes rendus des débats parlementaires constituent sans doute l'une

1. Martin (H.-Marie), « Variétés. Nouveau manuel de sténographie ou Art de suivre la parole en écrivant », *Le Constitutionnel : journal du commerce, politique et littéraire* (, 24 janv. 1867), URL : `https://gallica.bnf.fr/ark:/12148/bpt6k674517j` (visité le 12/08/2022).

2. *Ibid.*

des institutions démocratiques les moins connues du public et les moins étudiées par les spécialistes.[3] ».

Comment revaloriser les comptes rendus *in extenso* des débats parlementaires, un patrimoine méconnu et sous-utilisé, auprès des chercheurs et du grand public ? Comment favoriser la curiosité et l'envie d'explorer cette source riche et foisonnante ? Tels sont les enjeux auxquels le projet *Analyse sémantique et Graphes relationnels pour l'Ouverture et l'étude des Débats à l'Assemblée nationale* tentent de répondre. Ce projet, nommé plus communément AGODA, a pour objectif de créer une plateforme en ligne de consultation et d'exploration des débats parlementaires de la Chambre des députés de la III[e] République à partir des numérisations offertes par Gallica du *Journal officiel de la République française. Débats parlementaires. Chambre des députés : compte rendu in-extenso*[4]. Réunissant historiens et informaticiens, le projet AGODA vise à faciliter l'accès et l'exploitabilité de ces documents pour les chercheurs et le grand public grâce aux technologies numériques. Il repense la circulation du savoir en donnant aux internautes « de nouveaux moyens d'accéder à leurs observables[5] ».

Le projet AGODA souhaite traiter un corpus conséquent de comptes rendus, ceux publiés entre 1881 et 1940. Afin d'atteindre son objectif, il suit une méthodologie de traitement bien définie et accorde une grande importance à l'automatisation des tâches. La chaîne de traitement mise en place consiste à océriser les comptes rendus numérisés dans le but de rendre manipulables et exploitables les données textuelles. Ces dernières sont alors structurées et enrichies sémantiquement à l'aide d'un encodage en XML-TEI. Elles sont enfin stockées puis visualisées sur la plateforme de consultation *TEI Publisher*, celle-ci offrant diverses fonctionnalités d'exploitation. Cette chaîne de traitement vise l'éditorialisation[6] du corpus, autrement dit sa structuration, sa mise en accessibilité et sa mise en visibilité[7]. Elle contribue en cela à l'ouverture et l'exploitabilité des débats parlementaires.

3. Coniez (Hugo), « L'Invention du compte rendu intégral des débats en France (1789-1848) », *Parlement[s], Revue d'histoire politique* (, 2010), p. 146-158, URL : https://www.cairn.info/revue-parlements1-2010-2-page-146.htm (visité le 12/08/2022).

4. *Journal Officiel de La République Française - 70 Années Disponibles*, Gallica, URL : https://gallica.bnf.fr/ark:/12148/cb34378481r/date.r=Journal+officiel+de+la+republique+francaise+Lois+et+decrets.langFR (visité le 14/08/2022).

5. Longhi (Julien), « Humanités, numérique : des corpus au sens, du sens aux corpus », *Questions de communication*–31 (1er sept. 2017), URL : https://journals.openedition.org/questionsdecommunication/11039 (visité le 11/09/2022).

6. Concept défini par Bruno Bachimont dans son article « Nouvelles tendances applicatives : de l'indexation à l'éditorialisation », l'éditorialisation est un « processus consistant à enrôler des ressources pour les intégrer dans une nouvelle publication », Bachimont (Bruno), « Chapitre 1. Nouvelles tendances applicatives : de l'indexation à l éditorialisation », dans Patrick Gros, *L'indexation multimédia : description et recherche automatiques*, Paris, 2007, URL : https://docplayer.fr/127830894-Chapitre-1-nouvelles-tendances-applicatives-de-l-indexation-a-l-editorialisation.html.

7. Rosati (Marcello Vitali) et Sinatra (Michael E.), « Introduction », dans *Pratiques de l'édition numérique*, 2014, URL : http://parcoursnumeriques-pum.ca/1-pratiques/%E2%80%8Bhttps://www.parcoursnumeriques-pum.ca/1-pratiques/introduction.html (visité le 11/09/2022).

Ces différents enjeux ont été au cœur des missions qui m'ont été confiées. C'est ainsi mon intérêt général pour la recherche en histoire et les technologies numériques qui m'a conduite à choisir ce stage. Le projet AGODA était une opportunité pour moi d'allier ces deux domaines, et me permettait, par la même, d'appliquer les différents langages techniques appris au cours de l'année. Rattachée administrativement au Laboratoire de Recherche Historique Rhône-Alpes (LARHRA), j'ai effectué ce stage en présentiel dans les locaux d'Epitech, au sein du laboratoire Méthodes Numériques pour les Sciences de l'Humain et de la Société (MNSHS), sur une période de quatre mois, du 11 avril au 29 juillet et du 22 août au 28 août 2022. J'ai été encadrée par les deux coordinateurs du projet, Madame Marie Puren, enseignante-chercheuse à Epitech en histoire et humanités numériques et Monsieur Pierre Vernus, maître de conférences à l'Université Lumière Lyon 2 en histoire contemporaine. Durant ce stage, j'ai pu également bénéficier d'un appui régulier par les autres membres de l'équipe du projet et échanger avec de nombreuses personnes-ressources, selon les différents besoins des missions et lors de communications scientifiques.

Arrivée lors du processus d'océrisation et de post-correction des textes, j'ai été chargée de structurer et d'enrichir sémantiquement ces données océrisées en créant pour cela un modèle d'encodage XML-TEI, applicable automatiquement. Je devais donc réaliser :

- la modélisation d'un encodage en XML-TEI applicable aux particularités des comptes rendus *in extenso* ;
- la production d'un schéma pour valider l'encodage ;
- la rédaction d'une documentation pour expliciter et justifier les choix d'encodage effectués ;
- l'écriture de scripts, en langage de programmation python, pour automatiser l'application de l'encodage sur l'ensemble du corpus traité.

Ce présent mémoire rend compte des missions effectuées lors de ce stage [8]. Il permet d'illustrer les réflexions menées, les méthodes employées et les technologies utilisées. Nous présenterons dans un premier temps le contexte, en abordant les particularités de la source traitée, les objectifs du projet AGODA, et les premières étapes réalisées. Nous axerons ensuite le mémoire sur les problématiques propres aux missions du stage : nous verrons, d'une part, comment encoder de manière standard les comptes rendus *in extenso* des débats parlementaires, et d'autre part, nous évoquerons comment automatiser le processus d'encodage d'un corpus historique regroupant des textes nombreux et complexes.

8. Pour la rédaction de ce mémoire, j'utiliserai la première personne du singulier (je) pour mentionner les tâches que j'ai effectuées personnellement, et j'emploierai la première personne du pluriel (nous) pour la narration de l'écrit et les tâches effectuées en équipe.

Première partie

Les débats parlementaires au cœur du projet AGODA

Chapitre 1

Présentation de la source

1.1 Contextualisation historique

La Chambre des députés a joué un rôle central sous la III^e^ République. Proclamée le 4 septembre 1870 et fondée constitutionnellement par les lois constitutionnelles de 1875, elle prend un caractère clairement parlementaire à partir de 1879 une fois la conquête des institutions par les Républicains achevée. Dès lors, la Chambre des députés fut placée au cœur du système parlementaire instauré [1]. Le compte rendu *in extenso* des débats parlementaires de la Chambre des députés, publié au *Journal officiel*, relate les débats de cette institution et constituent une source majeure pour l'histoire du régime. Selon Benjamin Morelé : « Le compte rendu intégral des débats ne peut pas être compris sans appréhender son mode de production. [2] ». Il s'agira donc d'abord de comprendre comment celui-ci est produit par la Chambre des députés. Pour cela, nous étudierons le fonctionnement de cette institution afin d'éclairer le sens des comptes rendus, et nous expliciterons le contexte dans lequel il est apparu.

1.1.1 L'organisation des débats au regard du fonctionnement de la Chambre des députés [3]

Les comptes rendus des débats parlementaires relatent une organisation particulière qu'il est possible de comprendre au regard du fonctionnement politique de la Chambre des députés.

1. Mayeur (Jean Marie), *La Vie Politique Sous La Troisième République : 1870-1940*, Paris, 1984 (Points. Histoire).

2. Morel (Benjamin), « Ce que conte le compte rendu : l'institution d'un ordre parlementaire idéalisé », *Droit et société* (, 2018), p. 179-199, URL : `https://www.cairn.info/revue-droit-et-societe-2018-1-page-179.htm` (visité le 31/08/2022).

3. Les propos tenus dans cette partie reposent sur l'analyse issue du *Traité pratique de droit parlementaire* de Jules Poudra et Eugène Pierre. Ils sont inclus, de façon plus approfondie, au sein de la documentation du projet, Lebreton (Fanny), Puren (Marie) et Vernus (Pierre), *AGODA : Schéma TEI Pour Les Débats Parlementaires Français de La Chambre Des Députés*, juill. 2022, URL : `https://agoda-project.github.io/agoda_odd.html` (visité le 04/09/2022).

Les périodes législatives

Une législature correspond à la période durant laquelle l'Assemblée législative exerce ses pouvoirs [4]. Les députés de la Chambre étaient élus, par scrutin uninominal majoritaire à deux tours [5], pour une durée de quatre ans. Définie par le début et la fin du mandat de ces derniers, une législature dure donc quatre ans.

Une session parlementaire correspond à la période durant laquelle le Parlement se réunit pour délibérer. En effet, l'assemblée des députés ne siégeait pas de manière permanente tout au long de l'année. Deux types de sessions pouvaient être institués : la session dite ordinaire s'ouvrait « de plein droit, chaque année, le second mardi de janvier [...] sans décret de convocation [6] » et devait durer minimum cinq mois [7] ; la session dite extraordinaire, quant à elle, s'ouvrait à n'importe quel moment de l'année après la convocation de la Chambre des députés par le Président de la République et elle n'avait pas de date de clôture définie.

Une séance parlementaire s'inscrit au sein d'une législature divisée en plusieurs sessions parlementaires. Elle correspond à la période de la journée où les députés de la Chambre se réunissent pour délibérer.

Le Bureau définitif et ses attributions [8]

Pour assurer le bon fonctionnement des séances parlementaires de la Chambre des députés, un Bureau définitif était élu « chaque année, pour la durée de la session, et pour toute session extraordinaire qui aurait lieu avant la session ordinaire de l'année suivante [9] ». Celui-ci était composé d'un président, de quatre vice-présidents, de huit secrétaires et de trois questeurs [10].

Le président était chargé de diriger la séance, d'accorder la parole, de recevoir et transmettre les divers communications et projets, de diriger les votes, de constater les votes par assis et levée.

Les secrétaires, au nombre de quatre minimum lors des séances, avaient pour rôle de surveiller la rédaction du procès-verbal, en donner la lecture et le signer, d'inscrire les orateurs qui demandaient la parole, de dépouiller les scrutins publics, de procéder aux

4. Définition du mot « législature » issue du dictionnaire en ligne le Robert, `https://dictionnaire.lerobert.com/definition/legislature`.

5. Ce mode de scrutin était de mise lors de la III^{e} République, à l'exception des scrutins de la IV^{e} législature (1885-1889), et des XII^{e} et XIII^{e} législatures (1919-1927).

6. Poudra (Jules) et Pierre (Eugène), *Traité pratique de droit parlementaire*, 1878, URL : `https://gallica.bnf.fr/ark:/12148/bpt6k58651348` (visité le 12/08/2022), p. 211.

7. La Chambre pouvait être convoquée avant cette date si cela était jugé nécessaire par le Président de la République.

8. Propos issus du paragraphe « Attributions du Bureau définitif », Poudra et Pierre, *Traité pratique de droit parlementaire*, op. cit., p. 438-445.

9. *Ibid.*, p. 438.

10. *Ibid.*, p. 427.

appels nominaux, de prendre note des rappels à l'ordre, de constater les votes par assis et levé (avec le président), de constater le nombre de membres présents dans la salle.

Le déroulé des séances [11]

Une séance se déroule selon un protocole bien défini. L'ouverture, marquée par la déclaration du président « La séance est ouverte », avait lieu généralement à deux heures de l'après-midi. C'est à partir de ce moment là que la prise de parole était autorisée, et que les sténographes débutaient leur prise de notes. Le procès-verbal de la séance précédente était lu par l'un des secrétaires, puis une fois ce procès-verbal adopté, le président informait la Chambre des communications la concernant, et il appelait l'ordre du jour. Les délibérations pouvaient ensuite commencer.

Les projets de loi étaient élaborés par les commissions en amont et présentées à la Chambre sous forme de rapport. Ils étaient discutés et adoptés ou non selon le résultat des votes. Les orateurs n'avaient pas de limite précise pour la durée de leur discours. Il leur était demandé de présenter leur point de vue « sommairement », mais cette consigne était rarement respectée, car aucune sanction ni règle claire n'étaient mises en place. La lecture des discours écrits était autorisée et il était même possible de lire le discours d'un collègue absent, à condition d'en prendre la responsabilité et d'en approuver le principe. La prise de parole était réglementée et dirigée par le président de séance. Seul celui-ci avait le droit d'accorder la parole et d'autoriser l'orateur à monter à la tribune. Les demandes de prises de paroles étaient collectées et listées par les secrétaires. Ces listes étaient présentées au président au moment où le projet était discuté. Il était également possible de demander au président la parole à voix haute ou par inscription silencieuse lors du débat. Une fois qu'il avait pris la parole, un orateur ne pouvait être interrompu que par le président. L'ordre des prises de parole suivait l'ordre des listes des orateurs inscrits, en commençant par le premier orateur inscrit sur la liste des « contre ». La parole était ensuite donnée au premier orateur inscrit du côté « pour ». Seuls les ministres et les rapporteurs étaient dispensés de suivre l'ordre d'inscription et pouvaient obtenir la parole dès qu'ils la réclamaient. Il était possible aussi, dans certains cas, d'obtenir la parole de plein droit [12]. Une fois le vote ouvert, plus personne n'avait le droit de prendre la parole.

Le vote pouvait être effectué de deux manières différentes : le vote par assis et levé et au scrutin public. Pour le premier, le président faisait d'abord lever la partie de la Chambre « pour », puis il faisait lever ceux qui se prononçaient « contre ». Les votes étaient constatés par le président et les secrétaires. Pour le deuxième, les députés disposaient de deux bulletins avec leurs noms imprimés dessus. L'un était blanc et exprimait l'adoption ;

11. Propos issus des chapitres VI « Du débat » et XII « Des votations » , *Ibid.*, p. 597-610 et p. 686-729.

12. La parole pouvait être obtenue de plein droit pour les raisons suivantes : rappel au règlement, réclamation sur l'ordre du jour, priorité et question préalable, « fait personnel » soulevé au cours d'un discours ou d'un incident.

l'autre était bleu et exprimait la non-adoption. Les votants déposaient le bulletin choisi dans des urnes. Le résultat des votes était proclamé par le Président.

Une fois l'ordre du jour épuisé, ou si l'heure l'obligeait, ou encore si les députés semblaient fatigués, le président procédait à la clôture de la séance. Il demandait à la Chambre le jour, l'heure et l'ordre du jour de la discussion de la prochaine séance. Il pouvait faire ensuite des communications importantes à la Chambre avant de clôturer définitivement la séance. La prise de parole était alors interdite, les sténographes arrêtaient leurs prises de notes.

1.1.2 L'invention du compte rendu *in extenso* des débats parlementaires

Une publicisation croissante des débats parlementaires

Avant la Révolution française, « le secret était de règle dans les affaires publiques [13] ». Les assemblées délibéraient en huis clos, et très peu d'informations étaient publiées. Un tournant s'opéra à partir de la Révolution et tout au long du XIX^{e} siècle. Longtemps réprimée, l'intérêt pour les affaires politiques s'est développé abondamment à cette époque, plaçant les débats parlementaires au cœur des sujets de société. En parallèle de cet intérêt nouveau, les évolutions permises par la Révolution ont contribué à une publicisation croissante de la vie politique, qui s'est poursuivie au XIX^{e} siècle, notamment avec la transcription détaillée et la publication des débats parlementaires.

Reconnue depuis 1789, la publicité des débats a un double objectif. Selon la constitution du 3 septembre 1791 : « Les délibérations du Corps législatif seront publiques, et les procès-verbaux de ses séances seront imprimés [14] ». La publicité renvoie donc d'abord au droit d'assister aux séances parlementaires et ensuite à l'exposition au plus grand nombre des propos tenus grâce à la publication. De nombreux journalistes, appelés aussi rédacteurs du Parlement, pouvaient être présents lors des séances et relataient dans les journaux ce qui était dit. On voit alors apparaître une multitude de comptes rendus, publiés par les quotidiens. Ces derniers se sont imposés comme « l'instrument par excellence de la publicité politique [15] ».

La publicisation, au sens de publication, a connu deux tendances journalistiques. D'une part, il y avait les journaux qui tendaient vers l'officialisation des publications en essayant de restituer de façon fidèle les séances. Nous pouvons citer comme exemple le *Moniteur universel*, titre privé, mais considéré depuis 1799 comme un journal officiel. Ce dernier, rédigé au style direct, tentait tant bien que mal de reproduire fidèlement les

13. Coniez, « L'Invention du compte rendu intégral des débats en France (1789-1848) », op. cit.

14. Lavoinne (Yves), « Publicité des débats et espace public », *Études de communication. langages, information, médiations*–22 (1er déc. 1999), p. 115-132, URL : `https://journals.openedition.org/edc/2350` (visité le 12/04/2022).

15. *Ibid.*

détails des débats. D'autre part, il y avait les autres écrits journalistiques qui apportaient de façon affirmée une distance par la critique des propos relatés.

Cependant, les nombreuses publications des débats parlementaires ont été très vite contestées et contrôlées. La première moitié du XIX^{e} siècle a été marquée par un « contexte de conflits sur la liberté de la presse et d'exclusion politique du plus grand nombre [16] ». Ces conflits, plus ou moins prononcés selon les régimes, ont eu un fort impact sur la publicisation des débats. Les publications, aussi bien celles se voulant officielles, que les autres, étaient jugées imparfaites puisqu'elles n'étaient pas fiables. Les débats apparaissaient défigurés [17], au profit du caractère partisan des journalistes : « Selon l'orientation politique des journaux qui les publiaient, ils avantageaient tel ou tel des camps politiques en présence, en mutilant les interventions des orateurs adverses [18] ». Les publications pouvaient alors subir de nombreuses restrictions et codifications, allant même parfois jusqu'à la censure. Des délits de presse (amendes, emprisonnements) pouvaient être institués.

Vers une publicisation officielle des débats parlementaires

Au regard de cette situation, et pour pallier l'imperfection des publications, la publicisation des débats parlementaires s'est orientée de plus en plus vers l'officialisation de l'écrit, marquant alors une distinction plus prononcée et une concurrence plus importante entre les journaux officiels et la presse dite parlementaire. La publication officielle avait pour objectif de relayer des comptes rendus avec une authenticité garantie, et qui seul faisait foi. Plusieurs actions d'ordre technique et législatif ont été mises en place pour tendre vers cet objectif.

Afin d'obtenir des documents fiables, les débats devaient être retranscrits intégralement. Pour ce faire, diverses méthodes ont été utilisées. Par exemple, le *Journal logographique*, principal concurrent du *Moniteur* à la fin du XVIII^{e} siècle, avait mis en place une « Société logographique ». Situés dans une loge destinée aux rédacteurs des débats, les membres de cette société appliquaient une méthode de travail particulière, reposant sur l'alternance de la prise de notes [19]. Ces derniers essayaient de capter l'ensemble des détails, en vain. D'autres inventions ont eu lieu par la suite, jusqu'à ce que la sténographie parlementaire, initiée dès la fin du XVIII^{e} siècle, s'impose comme la pratique privilégiée. Cette dernière permettait de retranscrire la parole à l'aide d'une écriture abrégée, formée de signes, et donc d'accélérer la prise de note. Le *Sténographe des Chambres*, créé en 1831, fut considéré comme la « première publication entièrement fondée sur la sténographie [20] ».

16. Le Torrec (Virginie), « Aux frontières de la publicité parlementaire : les assemblées et leur visibilité mediatisée », *Réseaux* (, 2005), p. 181-208, URL : `https://www.cairn.info/revue-reseaux1-2005-1-2-page-181.htm` (visité le 12/08/2022).

17. Propos soutenus dans le paragraphe « Des comptes-rendus », Poudra et Pierre, *Traité pratique de droit parlementaire*, op. cit., p. 575-584

18. Coniez, « L'Invention du compte rendu intégral des débats en France (1789-1848) », op. cit.

19. *Ibid.*

20. *Ibid.*

Cette méthode, mise en application par d'autres journaux, a été complétée par une organisation beaucoup plus poussée et minutieuse du travail des sténographes. Leurs missions ont également été définies, codifiant ainsi la composition des comptes rendus.

Par ailleurs, le caractère officiel de l'écrit parlementaire s'est développé en parallèle d'une modification majeure : la publication officielle est passée « d'une activité économique privée dépendant du pouvoir » à « une activité étatique » [21]. Au départ, des avantages financiers (subvention de la Chambre des députés par exemple) et des avantages matériels (placement des rédacteurs dans des loges dédiées) pouvaient être alloués aux journaux dits « officiels ». Mais c'est en 1848 que le tournant s'opéra. L'Assemblée nationale adopta un décret instituant la création d'un corps sténographique d'État : « Le personnel du service sténographique sera attaché à l'Assemblée nationale, sous la direction du Bureau [22] ». Cette étape renoue d'ailleurs avec le second sens de « publiciser », signifiant le transfert des activités du domaine privé vers le domaine public ou de l'État [23].

Enfin, le caractère officiel de la transcription des débats parlementaires a été renforcé durant la III^{e} République. Les frontières entre les deux types de publicisation des débats (officiel et presse) ont été redéfinies concrètement grâce à la clarification du statut et des droits des journalistes parlementaires. La loi du 26 juin 1871 donna à ces derniers la possibilité de « publier leurs propres récits et appréciations des séances [24] ». Leur travail, mêlant analyse et critique, s'opposa donc aux publications officielles.

Le compte rendu *in extenso* des débats parlementaires du *Journal officiel de la République française*

La naissance des comptes rendus *in extenso* tels que nous les connaissons aujourd'hui a été longue. Comme nous venons de l'exposer, les publications officielles sont passées par les mains d'acteurs privés, leur donnant des formes diverses, avant d'être composées en 1848 par l'État, selon une codification particulière.

Publiés à cette époque au *Moniteur universel*, les comptes rendus *in extenso* de la Chambre des députés ont été distribués ensuite, à partir de 1869, par le *Journal officiel de la République française* [25]. Ce dernier publiait alors, sous une série unique, les lois et décrets, les débats, et les documents administratifs et il était considéré comme la principale publication officielle de la République française [26]. À partir de 1881, les débats

21. Le Torrec, « Aux frontières de la publicité parlementaire », op. cit.

22. Coniez, « L'Invention du compte rendu intégral des débats en France (1789-1848) », op. cit.

23. Définition du mot « publicisation » issue du dictionnaire La Toupie, `https://www.toupie.org/Dictionnaire/Publicisation.htm`.

24. Le Torrec, « Aux frontières de la publicité parlementaire », op. cit.

25. En 1869, le journal avait pour nom *Journal officiel de l'Empire français*, ce n'est qu'en 1870, qu'il prit le nom de *Journal officiel de la République française*.

26. Blum (Catherine), *Le Journal Officiel Dans Gallica (1869-1946)*, Le blog de Gallica, 19 janv. 2016, URL : `https://gallica.bnf.fr/blog/19012016/le-journal-officiel-dans-gallica-1869-1946?mode=desktop` (visité le 11/08/2022).

tenus à la Chambre des députés ont été consignés dans une édition du journal qui leur était spécifiquement consacrée, intitulée *Journal officiel de la République française. Débats parlementaires. Chambre des députés : compte rendu in-extenso*. En effet, le *Journal officiel* a été divisé en quatre collections distinctes, séparant alors les débats de la Chambre des députés, les débats du Sénat, les impressions de la Chambre des députés et les impressions du Sénat. Ce journal est imprimé entre 1869 et 1880 par une société privée, avant d'être repris par l'État en 1881 [27].

La publication des comptes rendus *in extenso* n'a pas changé depuis, les débats sont aujourd'hui toujours publiés au sein du *Journal officiel*.

1.2 Définition d'une typologie

Le compte rendu *in extenso* des débats parlementaires de la Chambre des députés, publié par le *Journal officiel*, est une typologie documentaire particulière. L'analyse de ses particularités va se diviser en deux étapes : il s'agira d'abord de définir le compte rendu *in extenso* en le distinguant du procès-verbal et du compte rendu analytique, nous développerons ensuite les missions des sténographes, afin de mieux appréhender son contenu.

1.2.1 Compte rendu *in extenso*, procès-verbal et compte rendu analytique : trois typologies distinctes [28]

« *In extenso* » signifiant « en entier », le compte rendu *in extenso* est une retranscription intégrale des débats. Il est le résultat d'une prise de notes précises lors des séances, puis d'une reprise au style écrit de ces dernières. Il retrace alors tous les éléments constitutifs de la séance, à la fois les discours tenus, les votes et leur résultat, mais aussi l'ouverture et la clôture, les interruptions et même l'atmosphère [29].

27. Journal officiel de la République française, dans *Wikipédia*, 2022, URL : `https://fr.wikipedia.org/w/index.php?title=Journal_officiel_de_la_R%C3%A9publique_fran%C3%A7aise&oldid=195403573` (visité le 13/08/2022).

28. Distinction issue du chapitre « Du procès-verbal et des comptes-rendus », Poudra et Pierre, *Traité pratique de droit parlementaire*, op. cit., p. 572-584

29. Pour consulter deux exemples de comptes rendus, se référer à l'annexe A.1.

Journal officiel du 27 Novembre 1889 CHAMBRE — SÉANCE DU 26 NOVEMBRE Session extraordinaire de 1889 175

CHAMBRE DES DÉPUTÉS

5e Législature. — Session extraordinaire de 1889.

COMPTE RENDU IN EXTENSO. — 16e SÉANCE

Séance du mardi 26 novembre.

SOMMAIRE

Procès-verbal : MM. Paul Déroulède, Georges Laguerre, Briens, Bizouard-Bert, Vernière.

Excuses et demandes de congé.

Communication d'une lettre de M. le ministre des finances relative à la nomination d'un membre de la commission de contrôle de la circulation monétaire.

Rapports d'élections par MM. Reybert, le comte Le Gonidec de Traissan, Delaunay.

Discussion des conclusions du rapport du 7e bureau sur l'élection de M. Dillon dans la 1re circonscription de l'arrondissement de Lorient (Morbihan). — Demande d'ajournement : MM. Georges Laguerre, le comte Lemercier, rapporteur. Rejet. — Discussion : MM. Cunéo d'Ornano, le comte Lemercier, rapporteur, le comte de Bouville-Maillefeu, Georges Laguerre. Adoption, au scrutin, des conclusions du bureau et annulation de l'élection.

Discussion des conclusions du rapport du 10e bureau sur l'élection de M. Arnault dans la 2e circonscription de Montauban (Tarn-et-Garonne) ; MM. Leygues, Arnault. — Demande, par M. Leygues, d'invalidation de l'élection. Adoption au scrutin.

Dépôt, par M. de Soland et plusieurs de ses collègues, d'une proposition de loi ayant pour objet de frapper d'un droit de douane, à leur entrée en France, les chanvres bruts ou préparés provenant de l'étranger.

Dépôt, par M. le marquis de La Ferronnays et plusieurs de ses collègues, d'une proposition de résolution tendant à la nomination d'une commission de onze membres à l'effet de reviser, s'il y a lieu, le régime légal et administratif en vigueur pour la protection du vignoble français contre le phylloxera.

Congés.

PRÉSIDENCE DE M. CHARLES FLOQUET

La séance est ouverte à deux heures.

M. Henri Lavertujon, *l'un des secrétaires,* donne lecture du procès-verbal de la séance d'hier.

M. Paul Déroulède. Je demande la parole.

M. le président. Vous avez la parole.

M. Paul Déroulède. Messieurs, j'ai demandé la parole pour faire une rectification au procès-verbal et pour m'expliquer au sujet du rappel à l'ordre qui m'a été infligé.

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... (*Bruit à gauche.*)

Voix au centre. Ce n'est pas une rectification.

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. (*Applaudissements sur quelques bancs à l'extrémité gauche de la salle. — Exclamations au centre et à gauche.*)

M. Gustave Rivet. La tolérance pour toutes les ignominies qui ont été commises pendant la période électorale.

CHAMBRE. — IN EXTENSO.

M. Paul Déroulède. Le procès-verbal attribue à notre groupe et à moi des paroles qu'aucun de nous n'a prononcées. Je n'ai jamais dit au sujet de cette validation ce que me fait dire le *Journal officiel* : « Oui, nous les validerons tous ! »

Le secrétaire sténographe qui m'a prêté ces paroles m'a sans doute pris pour quelqu'un des membres de cette majorité ministérielle qui peut tout ce qu'elle veut. (*Bruit.*) Nous ne sommes, nous, qu'une minorité...

Un membre au centre. Vous avez été la majorité hier !

M. Paul Déroulède. ...nous ne sommes, nous, qu'une minorité, et ce langage n'aurait eu aucun sens dans la bouche d'aucun de nous.

M. César Duval. Mais rien ne vous autorise à insulter la majorité !

M. Paul Déroulède. Je n'ai pas insulté, je n'insulte pas, de parti pris, la majorité ; au contraire je l'ai applaudie hier dans sa première manifestation de justice et de conscience. (*Très bien ! sur quelques bancs à l'extrémité gauche de la salle. — Bruit.*)

M. le président. Monsieur Déroulède, il faudrait vous renfermer dans les limites d'une rectification au procès-verbal. Vous pouvez dire en quoi vous le trouvez inexact, mais rien de plus.

M. Paul Déroulède. La rectification porte sur le point suivant: Je n'ai pas dit que nous validerions toutes les élections; j'ai dit, et je le répète : « Nous sommes pour la liberté contre les oppresseurs !... » (*Exclamations ironiques au centre et à gauche.*)

M. Jules Maigne. C'est pour cela que vous êtes le représentant de la dictature.

M. Paul Déroulède. « ...nous sommes pour la liberté pour tous. Vive la liberté ! »

Et la preuve que ce sont bien là les paroles que j'ai prononcées, c'est qu'elles sont ainsi rapportées dans le compte rendu analytique et ainsi rapportées dans la plupart des journaux du matin, et que le compte rendu sténographique seul donne une autre version.

Quant au rappel à l'ordre que ces paroles m'ont attiré, je ne proteste pas contre lui ; je l'accepte, au contraire, comme un symbole, comme un souvenir du centenaire de la Révolution française fêté par la République parlementaire. (*Exclamations en sens divers. — Applaudissements sur quelques bancs à l'extrémité gauche de la salle.*)

M. Laguerre. Le *Journal officiel* me porte comme m'étant abstenu dans le vote sur la validation de l'élection de M. de La Martinière; je déclare que j'étais à ma place et que j'ai déposé moi-même mon bulletin de vote en faveur de la validation. Je m'étonne de cette erreur, et je la regrette.

Je fais la même observation pour mon ami M. Martineau, qui a émis le même vote et qui est porté comme s'étant abstenu.

M. Briens. Messieurs, lorsque M. d'Estournel m'a interrompu pour demander quel était l'auteur de l'article que j'ai cité, on prête les paroles que j'ai prononcées à M. le rapporteur, tandis que c'est moi seul qui ai répondu à M. d'Estournel. Je demande que la rectification soit faite au procès-verbal.

M. Bizouard-Bert. Nous avons été retenus très tard dans nos bureaux pendant la séance d'hier; si j'avais été présent à la première partie de la séance, je déclare que j'aurais voté « pour » la proposition de M. Ferroul et pour celle de M. Maxime Lecomte, et cela dans un sentiment d'humanité.

M. Vernière. Au nom de M. Ménard-Dorian, retenu dans une commission, je suis chargé de déclarer qu'il s'est abstenu volontairement, hier, dans le vote relatif à l'élection de M. de La Martinière, tandis que dans le *Journal officiel* il est porté comme ayant voté « contre ».

M. le président. Il n'y a pas d'autres observations ?...

Le procès-verbal est adopté.

EXCUSES ET DEMANDES DE CONGÉ

M. le président. M. Le Cour s'excuse de ne pouvoir assister à la séance de ce jour.

MM. Girodet et Baïhaut s'excusent de ne pouvoir assister à la séance de ce jour, et demandent des congés.

Les demandes seront renvoyées à la commission des congés.

COMMUNICATION DU GOUVERNEMENT

M. le président. J'ai reçu de M. le ministre des finances la lettre suivante :

« Paris, le 11 novembre 1889.

« Monsieur le président,

« J'ai l'honneur de vous prier de vouloir bien proposer à la Chambre des députés de désigner un de ses membres pour faire partie, en remplacement de M. Fernand Faure, de la commission de contrôle de la circulation monétaire instituée par la loi du 31 juillet 1879.

« Cette commission devant, aux termes de l'article 5 de la loi précitée, se réunir à la fin de chaque année, il serait à désirer que la Chambre procédât à la nomination dont il s'agit dans l'une de ses premières séances.

« Agréez, monsieur le président, l'assurance de ma très haute considération,

« *Le ministre des finances,*

« Signé : ROUVIER. »

Acte est donné de la communication dont la Chambre vient d'entendre la lecture. Elle sera insérée au procès-verbal de la séance et déposée aux archives.

La Chambre ayant fixé à jeudi l'élection de deux membres de la commission supérieure de la caisse nationale des retraites pour la vieillesse et celle de deux membres de la commission de surveillance des Caisses d'amortissement et des dépôts et consignations, je vous proposerai, messieurs, de décider que l'élection d'un membre de la commission de contrôle de la circulation monétaire aura lieu samedi prochain.

Il n'y a pas d'opposition ?...

L'élection sera mise en tête de l'ordre du jour de samedi.

SUITE DE LA VÉRIFICATION DES POUVOIRS

M. le président. L'ordre du jour appelle la suite de la vérification des pouvoirs.

M. Reybert a la parole pour donner lecture d'un rapport sur une élection non contestée.

M. Reybert, *rapporteur.* — Département de la Corrèze, arrondissement de Tulle, 1re circonscription.

Les élections du 22 septembre ont donné les résultats suivants :

Electeurs inscrits, 17,650, dont le quart est de 4,263.

Nombre des votants, 12,322.

Bulletins blancs et nuls, à déduire, 133.

Suffrages exprimés, 12,189.

Ont obtenu :

MM. Borie, député sortant..... 7,508 voix
Vachal, ancien député... 4,748

M. Borie a été proclamé député comme ayant réuni un nombre de voix au moins égal à la majorité absolue des suffrages.

17

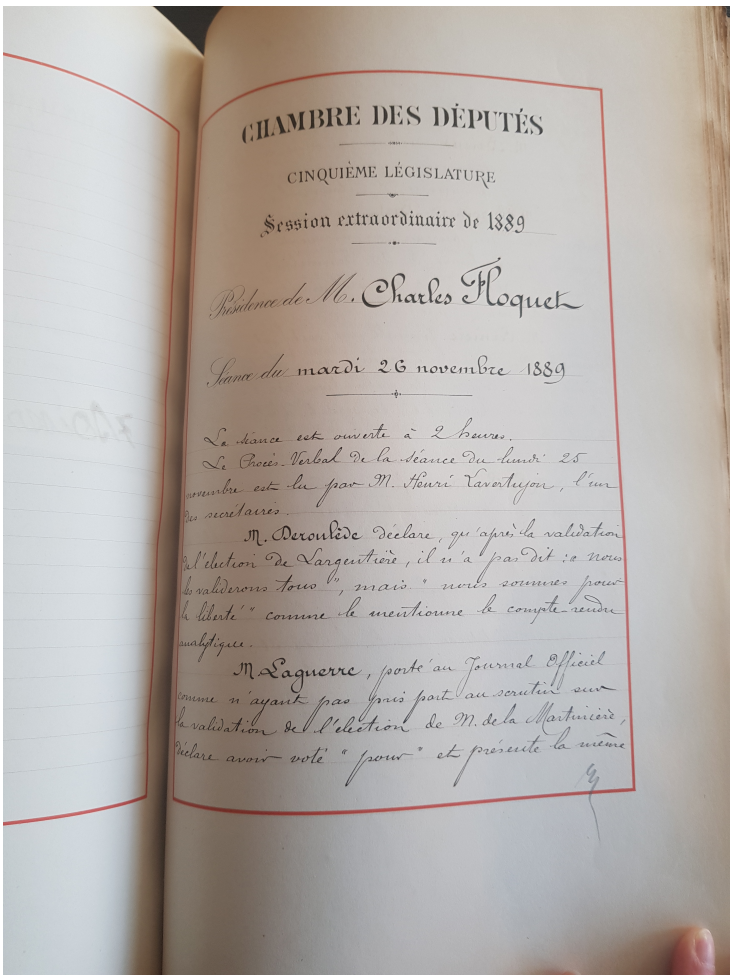
FIGURE 1.1 – Première page du compte rendu *in extenso* de la séance du 26 novembre 1889 (Gallica)

Il ne faut pas confondre cette typologie documentaire avec le procès-verbal qui est une autre façon de publier les débats. À l'origine, ce dernier était considéré comme le compte rendu officiel et était construit à l'aide du compte rendu analytique. Mais, contrairement au compte rendu *in extenso*, il résume les séances parlementaires. En effet, selon Eugène Pierre[30], le procès-verbal est le « constat des opérations et des votes de l'Assemblée, qui [contient] en outre le résumé des opinions de chacun des orateurs[31] ». Il est

30. Secrétaire général de la Chambre des députés de 1885 à 1925.

31. SAUDRAIS (Hélène), « Aux sources de la loi, les archives parlementaires (XIXe-XXe siècles) », *Revue française de droit constitutionnel* (, 2015), p. 165-175, URL : `https://www.cairn.info/revue-francaise-de-droit-constitutionnel-2015-1-page-165.htm` (visité le 12/08/2022).

constitué d'un résumé court, sans sommaire ni titres. Le détail des résultats des votes n'est pas précisé [32].



CHAMBRE DES DÉPUTÉS

CINQUIÈME LÉGISLATURE

Session extraordinaire de 1889

Présidence de M. Charles Floquet

Séance du mardi 26 novembre 1889

La séance est ouverte à 2 heures.

Le Procès-Verbal de la séance du lundi 25 novembre est lu par M. Henri Lavertujon, l'un des secrétaires.

M. Déroulède déclare, qu'après la validation de l'élection de Langeuttière, il n'a pas dit : « Nous les validerons tous », mais « nous sommes pour la liberté » comme le mentionne le compte rendu analytique.

M. Laguerre, porté au Journal Officiel comme n'ayant pas pris part au scrutin sur la validation de l'élection de M. de la Martinière, déclare avoir voté « pour » et présente la même

FIGURE 1.2 – Procès-verbal de la séance du 26 novembre 1889 (Archives nationales, cote C/I/446)

Le compte rendu analytique, quant à lui, est une typologie documentaire intermédiaire. Rédigé lors des séances, ce dernier n'était pas aussi complet que le compte rendu *in extenso*. Il était plus synthétique que celui-ci, mais ne constituait pas un résumé au même titre que le procès-verbal.

1.2.2 Le contenu des comptes rendus *in extenso* au regard des missions des sténographes

Le compte rendu *in extenso* est le résultat d'un travail complexe, mis en œuvre par les sténographes présents en séance. Ces derniers sont chargés de suivre la parole à l'écrit.

32. Ce constat a pu être illustré lors de notre consultation, aux Archives nationales, de plusieurs registres contenant les procès-verbaux de la III^e^ République, (cotes des registres consultés : C/I/446, C/I/448, C/I/450, C/I/452, C/I/454), se référer au dossier /PV en annexe pour voir un extrait A.2.

Ils doivent prendre en note de façon fidèle ce qu'ils entendent et voient. Ils font de leur travail un véritable « art d'écrire aussi vite que l'on parle [33] ».

Cependant, le compte rendu *in extenso* n'est pas une reproduction littérale de la séance. En effet, la mission des sténographes ne repose pas sur du mot à mot, mais sur une pratique rédactionnelle plus complexe. L'oralité est un style à part entière : le flux de paroles est très rapide, les propos véhiculés peuvent être déstructurés, ornés de tics langagiers, de répétitions, d'hésitations, d'incorrections. Retranscrire intégralement tous les détails de l'oral est un objectif difficilement réalisable pour les sténographes et par conséquent, tend à réduire la fidélité de leur travail. Il n'est, de plus, pas utile de préciser les ornements oraux puisqu'ils n'ajoutent aucun contenu véritable aux messages véhiculés, et peuvent même rendre la lecture des propos fatigante. Comme l'affirme Hugo Coniez : « l'écart est tel entre la langue écrite et la langue orale, même chez les meilleurs orateurs, même au XIX^{e} siècle, que le discours prononcé doit impérativement être rapproché des standards de l'écrit, faute de quoi le lecteur ne parvient pas à le suivre. [34] ». Ainsi, la mission des sténographes consiste à retravailler le style oral vers le style écrit afin qu'il soit compréhensible pour le lecteur, tout en restant fidèle à l'orateur. C'est « une œuvre intellectuelle de traduction et de remise en forme. [35] ». Cet aspect est d'ailleurs développé par Hippolyte Prévost au sein de l'article « Organisation de la sténographie officielle de l'Assemblée nationale » paru dans *Le Constitutionnel* du 19 juin 1848 [36].

Cette exigence [37] des sténographes permet de mieux comprendre le contenu du compte rendu *in extenso* : ce dernier « inexorablement exacte ne sera plus l'image de la parole, [il] en offrira la charge, la caricature [38] ».

1.3 L'état actuel de l'accessibilité et de l'exploitabilité de la source

Les comptes rendus *in extenso* des débats parlementaires de la Chambre des députés ont un rôle important. Ils « participe[ent] non seulement à la lisibilité de l'activité parle-

33. Prévost (Hippolyte), *Nouveau manuel de sténographie ou Art de suivre la parole en écrivant*, 4e éd. revue, augmentée et accompagnée de planches, 1834, URL : `https://gallica.bnf.fr/ark:/12148/bpt6k2991973` (visité le 12/08/2022).

34. Coniez, « L'Invention du compte rendu intégral des débats en France (1789-1848) », op. cit.

35. *Ibid.*

36. Prévost (Hippolyte), « Variétés. Organisation de la sténographie officielle de l'Assemblée nationale. » *Le Constitutionnel : journal du commerce, politique et littéraire* (, 19 juin 1848), URL : `https://gallica.bnf.fr/ark:/12148/bpt6k668244c` (visité le 13/08/2022).

37. Cette exigence du travail des sténographes est par ailleurs évoquée par C. Azéma, directeur du service du compte rendu intégral à l'Assemblée nationale, Rozenberg (Olivier) et Baudot (Pierre-Yves), « Entretien avec Claude Azéma, directeur du service du compte rendu intégral à l'Assemblée nationale », *Parlement[s], Revue d'histoire politique* (, 2010), p. 133-145, URL : `https://www.cairn.info/revue-parlements1-2010-2-page-133.htm` (visité le 12/08/2022). Elle est également développée dans l'analyse suivante : Morel, « Ce que conte le compte rendu », op. cit.

38. Prévost, « Le Constitutionnel », op. cit.

mentaire, mais aussi à sa légitimation et à la validité de ses actes. [Ils] nous raconte[nt] une histoire.[39] ». Toutefois, comme nous venons de le voir, cette « histoire » répond à ses propres logiques puisqu'elle dépend des missions des sténographes. Malgré cette limite, les comptes rendus sont une véritable « corne d'abondance pour la recherche[40] ». Ces derniers peuvent, en effet, intéresser de nombreux domaines tels que les sciences sociales et les sciences juridiques. Ils constituent en cela une source scientifique riche pour les chercheurs, mais aussi pour le grand public.

Cette source est consultable par tous. Cependant, comme le constate Hugo Coniez, elle reste trop peu exploitée. Nous allons mettre en avant d'abord les modes d'accès à cette source et nous expliciterons ensuite les causes de ce constat.

1.3.1 Une source accessible...

La conservation des comptes rendus

Les comptes rendus des débats parlementaires de la III^{e} République publiés au *Journal Officiel* sont conservés et librement communicables à tous. Il est possible d'y avoir accès de plusieurs façons et sous plusieurs formes.

Les comptes rendus sont d'abord consultables en version papier dans la majorité des services départementaux d'archives, aux Archives nationales, à la Bibliothèque nationale de France, et à la bibliothèque de l'Assemblée nationale et du Sénat. Ces versions papier peuvent être dupliquées parfois par des microfilms. Le lecteur peut donc consulter sur place, dans ces différents lieux de conservation, les numéros du *Journal officiel*. Au cours de notre visite aux Archives nationales (site de Pierrefitte-sur-Seine), nous avons pu, par exemple, explorer les comptes rendus de la Chambre des députés du *Journal officiel* en salle de lecture. Ils étaient en accès libre au lecteur, en version papier, sous forme de registres reliant les différents numéros du journal[41].

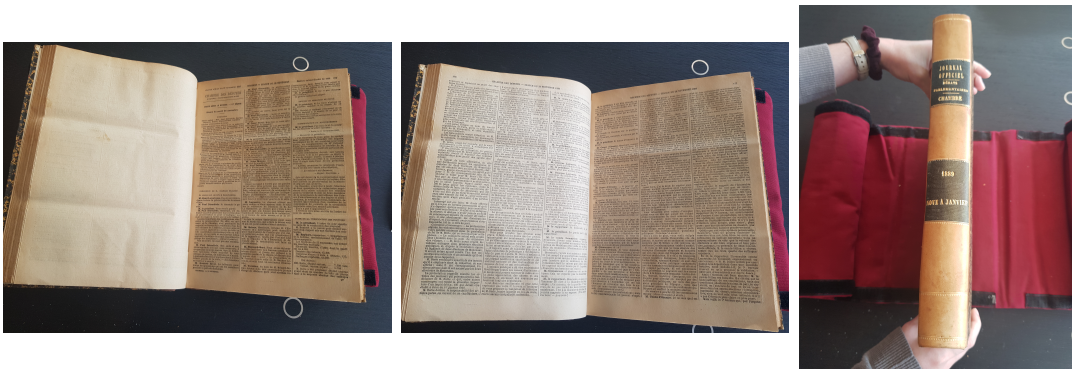


FIGURE 1.3 – Registre contenant les numéros du *Journal officiel* de novembre à janvier 1889

39. Morel, « Ce que conte le compte rendu », op. cit.
40. *Ibid.*
41. Se référer au dossier /CR de l'annexe A.2.

Par ailleurs, en plus de ce type de conservation, les comptes rendus des débats parlementaires de la III[e] République sont disponibles sur support numérique, grâce à leur mise en ligne [42] sur Gallica, la bibliothèque numérique de la Bibliothèque nationale de France [43]. Tous les débats parlementaires depuis la Révolution française ont été numérisés et mis en ligne sur Gallica entre 2009 et 2016 par la Bibliothèque nationale de France et les archives de l'Assemblée nationale. Ce processus de numérisation s'est inscrit « dans le cadre du programme français de numérisation des sciences juridiques [44] ». L'internaute peut donc consulter à distance ces numérisations, dans le corpus des « Essentiels du droit », au sein de la sous-catégorie « Législation et réglementation » [45] de Gallica.

FIGURE 1.4 – Compte rendu numérisé de la séance du 26 novembre 1889 sur Gallica

Une approche différente selon le format de la source

La conservation du *Journal officiel* se définit donc par différents modes d'accès, sur place ou à distance, et par différents supports. Ces supports offrent aux lecteurs différents

42. *Journal Officiel de La République Française - 70 Années Disponibles*, op. cit.

43. Gallica est une bibliothèque numérique accessible gratuitement sur l'internet. Elle donne un accès à tous types de documents (imprimés, manuscrits, documents sonores, documents iconographiques, cartes et plans, vidéos).

44. PUREN (Marie), BOURGEOIS (Nicolas), PELLET (Aurélien), VERNUS (Pierre) et LEBRETON (Fanny), « Between History and Natural Language Processing : Study, Enrichment and Online Publication of French Parliamentary Debates of the Early Third Republic (1881-1899) », dans *ParlaCLARIN III at LREC2022 - Workshop on Creating, Enriching and Using Parliamentary Corpora*, Marseille, France, 2022, URL : `https://hal.archives-ouvertes.fr/hal-03623351` (visité le 09/08/2022).

45. *Droit, économie, politique et presse dans Gallica*, BnF - Site institutionnel, URL : `https://www.bnf.fr/fr/droit-economie-politique-et-presse-dans-gallica` (visité le 11/08/2022).

modes d'approche de la source. Ils ont des enjeux scientifiques qui leur sont propres, et mettent en avant des fonctionnalités spécifiques. D'abord, le support papier correspond à la source primaire du document. Sa conservation matérielle permet de garder une trace de la matérialité du journal, en plus du contenu. L'aspect physique du document peut être en effet une information intéressante pour le chercheur. De plus, au contact du support papier, le lecteur des débats peut avoir une expérience de lecture diverse. Selon son envie et son besoin, il peut feuilleter rapidement les registres ou parcourir de façon précise les numéros.

Au contraire, la numérisation permet d'obtenir le fac-similé numérique de la version papier. Ce format de conservation garantit l'intégrité de l'original puisqu'il le reproduit à l'identique, mais sous une forme numérique. La numérisation est aussi un moyen de préservation du journal. En effet, la communication ne passe plus par la manipulation du support physique du document, mais par une consultation dématérialisée. En outre, les numérisations peuvent être mises en ligne et faciliter alors l'accessibilité du *Journal officiel*. Par ces différents enjeux, la numérisation entend répondre au problème des institutions de conservation dont le rôle est à la fois de conserver et de donner accès aux documents. Ce support permet également d'offrir à l'internaute différentes approches de lectures grâce à la mise à disposition de différentes fonctionnalités, ces dernières étant spécifiques à l'interface de diffusion.

Les numérisations des comptes rendus disponibles sur Gallica sont réparties année par année et mois par mois [46]. Il est possible de visualiser les images numériques dans l'interface même, de les télécharger au format JPEG ou PDF, ou encore de les imprimer. Gallica met à disposition des métadonnées [47] donnant un ensemble d'informations sur le numéro consulté, mais aussi plusieurs outils de lecture, notamment le zoom, et le choix du mode d'affichage des pages.

En plus de la numérisation en mode image du journal, la Bibliothèque nationale de France a réalisé une numérisation en mode texte des documents. Pour extraire le texte des images, elle a utilisé le logiciel de transcription automatique ABBY FineReader. Ce mode texte est très régulièrement combiné à celui du mode image puisqu'il permet de compléter ses fonctionnalités [48]. En effet, en donnant accès au texte océrisé, Gallica rend possibles l'indexation et la manipulation du contenu des débats. L'internaute peut effectivement récupérer par copier-coller des fragments de textes, ou encore utiliser le moteur de recherche afin d'accéder au contenu souhaité. Il peut également télécharger au

46. *Journal Officiel de La République Française - 70 Années Disponibles*, op. cit.

47. « Data about data are referred to as metadata. » (Les données des données sont des métadonnées). Définition des métadonnées issue de : Calderan (Lisette), Hidoine (Bernard) et Millet (Jacques), *Métadonnées : Mutations et Perspectives : Séminaire INRIA, 29 Septembre-3 Octobre 2008, Dijon*, Paris, 2008 (Sciences et Techniques de l'information).

48. Dacos (Marin) et Mounier (Pierre), *L'édition électronique*, Paris, 2010 (Repères, 549), p. 49.

format TXT l'ensemble du texte. Ces modes de visualisation offerts par Gallica permettent à l'internaute d'interagir de façon réactive avec le document, selon ses besoins.

1.3.2 ...mais difficilement exploitable

Comme nous venons de le voir, les débats parlementaires sont une source accessible. Un certain nombre de travaux scientifiques ont utilisé et mis en avant cette source. Par exemple les travaux de Bruno Marnot portant sur les ingénieurs au Parlement de la III^{e} République [49], ou encore ceux de Claire Lemercier concernant le travail des enfants dans l'industrie [50]. Mais malgré cette accessibilité des débats parlementaires, en version papier et en format numérique, ils restent encore trop peu étudiés.

Cela est lié aux caractéristiques de la source et aux moyens mis à disposition pour les exploiter. En effet, bien que les débats parlementaires soient facilement accessibles, les différents modes d'archivage ne permettent pas de les exploiter pleinement. La recherche au sein des débats est difficile, car la source est très dense. La consultation des comptes rendus nécessite de savoir en amont le sujet recherché, obligeant, par la même, le chercheur à avoir une bonne connaissance de la source. En effet, s'il souhaite accéder aux débats portant sur une loi en particulier, il doit connaître en amont la date à laquelle les discussions ont eu lieu. La version numérique des débats offre tout de même la recherche en texte intégral pouvant s'appliquer sur différents niveaux (recherche sur l'ensemble des numéros, les numéros selon l'année, les numéros selon le mois, le numéro en question). Elle permet de trouver, sans connaissances développées et à l'aide d'un mot clef, une information précise. Mais le résultat de la recherche ne permet de trouver que des éléments ponctuels, et n'est pas toujours concluant, car le texte océrisé contient des fautes. En plus de cet outil lacunaire, l'étude des sujets, dont certains peuvent être débattus sur plusieurs années, est difficile voire impossible, car déterminer le corpus dans lequel ils se développent reste une tâche laborieuse. L'étude de la source est alors limitée puisqu'il n'est pas possible de l'analyser pleinement et facilement. Les comptes rendus des débats parlementaires constituent ainsi une source difficilement exploitable.

49. Marnot (Bruno), *Les ingénieurs au Parlement sous la IIIe République*, 2000, URL : `https://gallica.bnf.fr/ark:/12148/bpt6k3373008g` (visité le 13/08/2022).

50. Lemercier (Claire), « Un catholique libéral dans le débat parlementaire sur le travail des enfants dans l'industrie (1840) », *Parlement[s], Revue d'histoire politique* (, 2021), p. 195-206, URL : `https://www.cairn.info/revue-parlements-2021-1-page-195.htm` (visité le 13/08/2022).

Chapitre 2

Analyse sémantique et Graphes relationnels pour l'Ouverture des Débats à l'Assemblée : présentation du projet

2.1 Les objectifs du projet

Le projet *Analyse sémantique et Graphes relationnels pour l'Ouverture et l'étude des Débats à l'Assemblée nationale* entend répondre aux problématiques que nous venons d'exposer. Nous allons voir comment il s'inscrit au cœur de ces dernières en présentant ses objectifs scientifiques et techniques.

2.1.1 Les origines

AGODA se situe dans un mouvement visant à améliorer la connaissance et l'exploitation des débats parlementaires ; il s'inscrit ainsi dans la lignée de plusieurs projets partageant des objectifs similaires. On peut citer à titre d'exemple la mise en ligne du Hansard britannique [1]. L'interface Web permet ainsi aux utilisateurs d'explorer les comptes rendus des débats parlementaires du Parlement britannique [2]. Nous pouvons mentionner également les projets ParlaClarin et ParlaMint. Ces deux projets proposent de produire des corpus multilingues, encodés en XML-TEI, de débats parlementaires, afin de faciliter leur échange et leur réutilisation. À partir de ces corpus annotés, plusieurs analyses ont

1. Il s'agit du nom donné aux retranscriptions des débats dans les gouvernements de type Westminster, par exemple au Royaume-Uni, au Canada ou encore à Singapour.
2. Disponible à l'adresse suivante : `https://hansard.parliament.uk/`.

pu être réalisées [3]. L'encodage des débats parlementaires a également été appliqué par Naomie Truan dans le cadre de sa thèse. Elle a proposé une annotation linguistique en XML-TEI sur trois corpus de débats parlementaires, ceux de la Chambre des communes britannique, ceux du Bundestag allemand et ceux de l'Assemblée nationale française [4].

Ces projets sont des réponses aux problématiques liées aux débats parlementaires, puisqu'ils proposent des solutions afin de revaloriser l'étude de cette source auprès des chercheurs et du grand public. Pour cela, ils se sont tournés vers les technologies numériques, et ont affilié la problématique des débats parlementaires au vaste champ des humanités numériques. Ce domaine, synonyme de redécouverte des sources selon Pierre Mounier [5], est « un dialogue interdisciplinaire sur la dimension numérique des recherches en sciences humaines et sociales, au niveau des outils, des méthodes, des objets d'études et des modes de communication. [6] ». En effet, les outils numériques permettent d'offrir « de nouveaux modes de lectures des sources » et permettent également de « modifier l'amplitude et la diversité des sources sur lesquelles [les chercheurs] appuient leur argumentation, permettant de faire surgir des questions nouvelles » [7]. Face à ces possibilités novatrices liées au numérique, les projets évoqués ci-dessus ont vu, par l'utilisation de ces outils, le moyen de développer de nouvelles visualisations et méthodes d'exploitations des débats, donnant ainsi de nouvelles perspectives de recherche et entraînant en cela un nouvel engouement pour les débats.

2.1.2 Les objectifs scientifiques

Le projet AGODA axe ses objectifs au carrefour de ces différents projets [8]. Il veut revaloriser les débats parlementaires en favorisant l'ouverture et l'étude de cette source. Il aspire à :

3. Diwersy (Sascha) et Luxardo (Giancarlo), « Querying a Large Annotated Corpus of Parliamentary Debates », dans *Proceedings of the Second ParlaCLARIN Workshop*, Marseille, France, 2020, p. 75-79, URL : https://aclanthology.org/2020.parlaclarin-1.13 (visité le 10/03/2022) et Diwersy (Sascha), Frontini (Francesca) et Luxardo (Giancarlo), « The Parliamentary Debates as a Resource for the Textometric Study of the French Political Discourse », dans *Proceedings of the ParlaCLARIN@LREC2018 Workshop*, Miyazaki, Japan, 2018, URL : https://hal.archives-ouvertes.fr/hal-01832649 (visité le 10/03/2022).

4. Truan (Naomi), *"Who Are You Talking About ?". The Pragmatics of Third-Person Referring Expressions : A Contrastive Corpus-Based Study of British, German, and French Parliamentary Debates*, These de doctorat, Sorbonne université, 2019, URL : http://www.theses.fr/2019SORUL014 (visité le 15/03/2022).

5. Mounier (Pierre), *Les humanités numériques : Une histoire critique*, Paris, 2018 (Interventions), URL : http://books.openedition.org/editionsmsh/12006 (visité le 11/08/2022).

6. Dacos (Marin) et Mounier (Pierre), *Humanités Numériques : État Des Lieux et Positionnement de La Recherche Française Dans Le Contexte International*, Research Report, Institut français, 2015, p. 9782354761097, URL : https://hal.archives-ouvertes.fr/hal-01228945 (visité le 11/08/2022).

7. Id. *L'édition électronique*, op. cit.

8. Puren, *et al.*, « Between History and Natural Language Processing », op. cit.

— donner plus facilement accès aux retranscriptions anciennes des débats parlementaires de la III[e] République (1881-1940) ;
— faciliter la recherche dans ce corpus ;
— permettre la constitution de sous-corpus ;
— offrir de nouveaux modes de visualisation des documents.

Afin de répondre à ces enjeux, AGODA a plusieurs objectifs. D'une part, le projet souhaite créer une plateforme de consultation des débats parlementaires. Cette dernière a pour rôle de faciliter l'accès aux débats en proposant des outils variés d'exploration tels que la recherche plein-texte, la navigation, la sélection de sous-corpus homogènes, l'identification des orateurs, etc. La plateforme doit donner les moyens aux utilisateurs de combiner plusieurs stratégies de lecture des textes, en proposant en plus d'une lecture « proche », des méthodes de « lecture distante » (modélisation de sujets et analyse de réseaux par exemple) [9].

D'autre part, AGODA a pour objectif de produire un corpus de données textuelles structurées et sémantiquement enrichies à partir de ces débats numérisés. L'annotation permet de mettre en avant la structure des comptes rendus (annotation structurelle) et de repérer les informations importantes des débats, qu'il serait intéressant d'extraire et de comparer (annotation sémantique). L'annotation a pour exigence, aussi, de s'inscrire dans une perspective de *linked data* en identifiant, lorsque cela est possible, les entités nommées (orateurs, lieux, organisations) avec les URI fournis par la BnF. Annoter les débats permet le développement de leur exploitabilité et de leur réutilisabilité.

Enfin, le projet AGODA souhaite contribuer à la conception d'un *workflow* adapté à l'analyse de grands corpus de documents historiques, en mettant en place une chaîne de traitement spécifique, faisable et réutilisable par d'autres projets. Ce *workflow* entend mettre l'accent aussi, plus largement, sur la production de données FAIR [10] en utilisant et respectant les langages standards de l'informatique, en documentant le travail effectué, et en rendant librement accessible le code source des outils développés.

2.2 Un travail à plusieurs mains

2.2.1 Le DataLab de la bnf

Le DataLab est un service de la BnF destiné aux chercheurs et partenaire de la très grande infrastructure de recherche (TGIR) HumaNum. Il a pour mission « d'encourager l'effort de recherche sur des thématiques numériques innovantes » à partir des collections numériques de la BnF et « de faire émerger des projets dans un environnement de recherche adapté » [11]. Pour cela, il accompagne les chercheurs sur les différentes étapes de leur projet

9. *Ibid.*
10. Acronyme pour *Findable, Accessible, Interoperable, Reusable.*
11. Site web disponible à l'adresse suivante : `https://www.bnf.fr/fr/bnf-datalab`.

et favorise la mise en commun et le partage des compétences et savoir-faire entre les équipes. Il offre, en plus d'un accompagnement scientifique et technique, un financement pour certains projets.

Le projet AGODA fait partie des cinq projets-pilotes sélectionnés suite au premier appel à projets lancé conjointement par la BnF et HumaNum en juin 2021. Il s'inscrit dans l'axe 3 de l'appel à projets : « Gallica et les collections numérisées ». Il est financé par le DataLab pour une durée d'un an et bénéficie de son accompagnement. Le suivi est effectué notamment par la coordinatrice BnF du DataLab, Marie Carlin.

2.2.2 Les équipes du projet

Le projet AGODA est le fruit d'une collaboration entre plusieurs laboratoires. Il est mené par différentes équipes aux compétences variées. Il associe le Laboratoire de Recherche Historique Rhône-Alpes (LARHRA), le laboratoire Méthodes Numériques pour les Sciences de l'Humain et de la Société (MNSHS) et l'Institut national de recherche en sciences et technologies du numérique (Inria-ALMAnaCH). Le projet AGODA convoque également d'autres équipes qui lui viennent en appui sur différents sujets centraux.

LARHRA

Le Laboratoire de Recherche Historique Rhône-Alpes (LARHRA) (UMR 5190) est une Unité Mixte de Recherche du CNRS, sous la tutelle des Universités Lumière-Lyon 2, Jean Moulin-Lyon 3, Grenoble-Alpes et de l'ENS de Lyon. Laboratoire généraliste, il aborde diverses thématiques en histoire et en histoire de l'art pour les périodes moderne et contemporaine. Il a également développé des compétences en histoire numérique au sein de son Axe de recherche en histoire numérique (ARHN). Il couvre des aires géographiques variées et partage une approche centrée sur les acteurs en prenant en compte toutes les dimensions du social [12]. Pierre Vernus, responsable et coordinateur du projet AGODA, est un des membres de ce laboratoire. Il est spécialisé en histoire économique et sociale de la période contemporaine et en humanités numériques.

MNSHS

Le laboratoire Méthodes Numériques pour les Sciences de l'Humain et de la Société (MNSHS) est le laboratoire de recherche de l'*European Institute of Technology* (Epitech). Il a pour objectif « d'étudier l'interaction entre facteurs technologiques et facteurs sociaux dans les réponses aux défis écologiques, économiques et politiques à venir [13] ». Plusieurs membres du projet sont issus de ce laboratoire. Marie Puren, dont les travaux de recherche

12. Site web disponible à l'adresse suivante : `http://larhra.ish-lyon.cnrs.fr/`.
13. Site web disponible à l'adresse suivante : `https://recherche.epitech.eu/`.

portent sur l'histoire littéraire et politique de la période contemporaine et sur les humanités numériques, est la coordinatrice du projet AGODA. Monsieur Nicolas Bourgois, docteur en Informatique et directeur du laboratoire et Monsieur Aurélien Pellet ingénieur de recherche, sont chargés des tâches d'océrisation, de post-correction et de fouille de texte.

Inria et ALMAnaCH

L'Institut national de recherche en sciences et technologies du numérique (Inria) est un Établissement Public à caractère Scientifique et Technologique (EPST) placé sous la double tutelle du ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation et du ministère de l'Économie et des Finances. Ses missions se situent au cœur l'innovation technologique [14]. Une des équipes projet de cet institut participe au projet AGODA, il s'agit de l'équipe ALMAnaCH dont l'acronyme signifie *Automatic Language Modelling and Analysis & Computational Humanities* ou Modélisation et analyse automatique du langage et humanités computationnelles. Composée d'informaticiens, cette équipe a pour sujet d'étude le traitement automatique des langues (TAL), sujet central des domaines de l'Intelligence Artificielle et des Humanités Numériques [15]. Éric de la Clergerie, chercheur en TAL et membre de l'équipe ALMAnaCH, est rattaché au projet AGODA. Julien Martin, développeur freelance employé par l'Inria, à l'origine notamment du développement de *web components* pour la plateforme TEI Publisher, est également associé au projet. Ces deux membres contribuent aux réflexions et à la mise en application de l'éditorialisation du corpus.

Les équipes en appui

Le projet AGODA bénéficie de l'appui du Laboratoire de Recherche et Développement (LRDE) de l'École pour l'informatique et les techniques avancées (Epita). Ce laboratoire réunit docteurs et doctorants en informatique, mathématiques ou traitement du signal. Il centre une partie de ses recherches sur le traitement d'images et la reconnaissance des formes [16]. Joseph Chazalon et Edwin Carlinet, très intéressés par ces deux sujets, collaborent avec l'équipe du projet sur les questions d'océrisation.

Le projet AGODA est également suivi par la Division des Archives et de l'Histoire parlementaire de l'Assemblée nationale. Chargé de la conservation et de la valorisation des archives de l'Assemblée nationale, le directeur de cette division, Monsieur Vincent Tocanne, aide l'équipe du projet en apportant son savoir historique et technique.

14. Site web disponible à l'adresse suivante : `https://www.inria.fr/fr`.
15. Site web disponible à l'adresse suivante : `https://www.inria.fr/fr/almanach`.
16. Site web disponible à l'adresse suivante : `https://www.lrde.epita.fr/wiki/Home`.

2.2.3 La gestion du projet

Comme nous venons de le voir, le projet AGODA associe de nombreuses personnes issues de laboratoires différents. Il favorise l'échange et le partage entre chercheurs, informaticiens et avec les institutions de conservation. Cette pluralité est une réelle richesse et apporte des avantages considérables pour la réalisation du projet. En effet, les différentes missions d'AGODA impliquent des compétences variées et spécifiques, celles-ci pouvant être exécutées par les différents profils des membres de l'équipe.

Cependant, cette collaboration multiple a engendré quelques difficultés dans la gestion du projet et a nécessité des aménagements techniques. En effet, l'éloignement spatial entre ses membres nous a poussés à développer une bonne coordination au sein de l'équipe. Nous avons défini une organisation précise avec une répartition des tâches en amont et un suivi régulier des avancées de chacun. Pour cela, nous avons utilisé les outils numériques dédiés tels que des plateformes de réunions virtuelles (Zoom [17], Teams [18]) pour effectuer des points réguliers, GitHub [19] afin de stocker et suivre les avancées du projet, et Google Drive [20] pour réaliser des tâches collaboratives comme la rédaction de la documentation. La contrainte de l'éloignement spatial a eu un impact aussi sur l'avancée du projet. Bien que nous échangions très régulièrement, l'échange à distance nous a fait perdre en efficacité, allongeant parfois les délais de prises de décisions et de réalisation des tâches. Nous avons réussi, toutefois, à nous réunir en présentiel quelques fois, ce qui nous a permis de faciliter nos échanges et d'avancer plus rapidement.

2.3 Le *workflow* du projet

La mise en œuvre des objectifs du projet AGODA nécessite la conception d'un *workflow* adapté à la production et à l'analyse de grands corpus de documents historiques. Nous exposerons d'abord les principales étapes de la chaîne de traitement, puis nous expliciterons l'approche du projet comme preuve de concept.

2.3.1 Une chaîne de traitement spécifique aux grands corpus de documents historiques

Le projet AGODA repose sur une chaîne de traitement, c'est-à-dire sur une suite d'opérations successives permettant le traitement de documents dans le but de réaliser les objectifs fixés. Sa chaîne de traitement est spécifique au traitement de grands corpus de

17. Plateforme disponible à l'adresse suivante : `https://explore.zoom.us/fr/products/meetings/`.

18. Plateforme disponible à l'adresse suivante : `https://www.microsoft.com/fr-fr/microsoft-teams/group-chat-software`.

19. *GitHub*, GitHub, URL : `https://github.com` (visité le 14/08/2022).

20. Plateforme disponible à l'adresse suivante : `https://www.google.com/intl/fr/drive/`.

documents historiques. En effet, celle-ci s'inspire d'une autre chaîne de traitement, mise en place dans le cadre du projet ANR TIME-US [21] (2018-2021). Ce projet avait pour objectif de traiter un grand corpus de documents historiques hétérogènes (sources juridiques, sources provenant du monde de l'entreprise, archives économiques, documents de presse) afin d'en faciliter l'accès, la recherche et la visualisation. Ce projet avait donc mis en place une chaîne de traitement spécifique [22] à ses besoins, à savoir collecter et numériser les sources, extraire les textes à l'aide de l'océrisation, structurer automatiquement les données selon les normes TEI, les stocker dans une base de données et les éditorialiser et publier en ligne [23]. Le projet AGODA partage des objectifs communs, c'est pourquoi il s'est inspiré des différentes étapes de la chaîne de traitement du projet TIME-US, tout en l'adaptant selon ses besoins spécifiques.

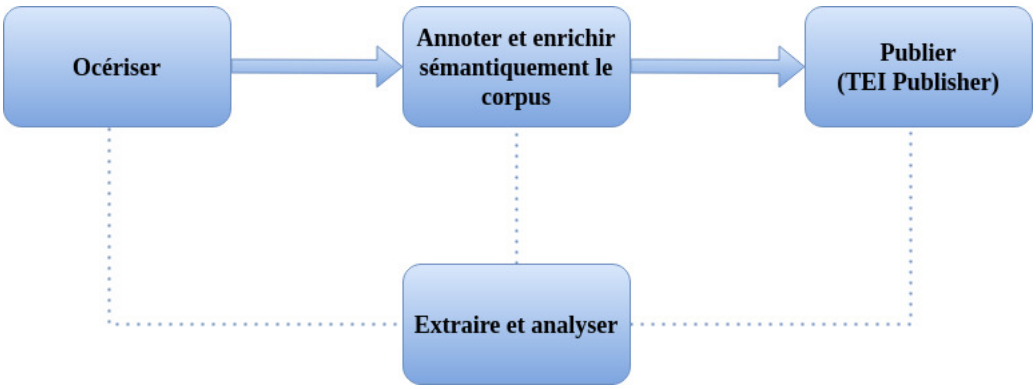


FIGURE 2.1 – Illustration de la chaîne de traitement

La chaîne de traitement du projet AGODA se divise en quatre étapes principales :

— Utilisant les numérisations des comptes rendus du *Journal officiel* disponibles sur Gallica, la première étape du projet consiste à extraire le texte de ces images numérisées à l'aide de l'océrisation. Le but est d'obtenir les textes avec le moins de fautes possible afin de pouvoir les exploiter dans la suite du processus, et de les valoriser *in fine*. Cette exigence nécessite un long travail d'essais et de choix d'outils puis de corrections ;

— La deuxième étape repose sur l'annotation et l'enrichissement sémantique des données issues de l'océrisation. Cette mission comprend deux temps avec d'abord la modélisation des choix d'encodage et des choix d'outils techniques, puis l'annotation concrète, c'est-à-dire l'application des balises choisies sur les textes océrisés. Cette application doit être automatisée puisqu'en effet, AGODA traite un corpus textuel massif ;

21. Le projet ANR TIME-US, ou Time usage, porte sur les rémunérations et usages du temps des femmes et des hommes dans l'industrie textile en France de la fin XVII^{e} siècle au début du XX^{e} siècle.

22. Travaux disponibles sur le dépôt GitLab du projet : *Time Us · GitLab*, GitLab, URL : `https://gitlab.inria.fr/almanach/time-us` (visité le 14/08/2022).

23. Visualisation du résultat disponible sur la page web suivante : `http://timeusage.paris.inria.fr/exist/apps/timeus-corpus/index.html`.

— Une fois les données structurées et enrichies, la troisième étape consiste à stocker les fichiers annotés en TEI dans une base de données eXist-db [24] puis à les publier via la plateforme TEI Publisher [25]. Cette dernière est capable de transformer les données sources issues de la base de données XML, en pages web HTML pour la publication. Les débats parlementaires sont ainsi mis en ligne, sous forme d'édition numérique, intégrés dans un contexte applicatif puisque TEI Publisher propose, en effet, des fonctionnalités de navigation, de recherche plein texte, et d'affichage des facsimilés. Il permet également d'extraire les données pour pouvoir les exploiter ;

— La dernière étape porte donc sur l'analyse des données traitées. Cette dernière peut être effectuée à l'issue de la publication des données. Mais elle peut également être intégrée au sein des étapes précédentes. Cela implique deux objectifs :

- Les analyses qui sont effectuées afin de mieux connaître les données, pour les valoriser ensuite selon de nouveaux modes de visualisation au sein de la dernière étape d'éditorialisation des données ;
- Les analyses futures réalisées par les utilisateurs de la plateforme de consultation à venir. Ces dernières pourront être effectuées grâce à l'extraction des données issues de cette plateforme à l'aide d'une API.

Cette chaîne de traitement entend produire des données trouvables, accessibles, interopérables et réutilisables (principes FAIR). Chacune de ces étapes tient compte donc un ensemble de tâches essentielles : le respect des standards informatiques, la production d'une documentation, et l'accessibilité du travail effectué.

2.3.2 Une approche comprise comme preuve de concept

Le projet AGODA s'inscrit dans une approche « preuve de concept », car il cherche à démontrer l'applicabilité, la faisabilité et la viabilité de la chaîne de traitement élaborée. Après avoir conceptualisé les étapes de cette dernière, l'objectif d'AGODA est de les mettre en application pour les tester. Le projet souhaite prouver qu'il est possible d'atteindre les objectifs fixés.

Afin de tester leurs hypothèses, les membres de l'équipe du projet ont fait le choix de limiter la quantité des documents à traiter en ne prenant en charge, dans un premier temps, qu'un sous-corpus des comptes-rendus, ceux publiés lors de la V^{e} législature (1889-1893) de la III^{e} République. Ce sous-corpus représentait déjà un ensemble de documents conséquents avec 10418 images numérisées. Le choix de ce sous-corpus reposait aussi sur

24. eXist-db est un système de gestion de base de données *open source*, basé sur le XML, disponible à l'adresse suivante : `http://exist-db.org/exist/apps/homepage/index.html`.

25. Plateforme permettant de publier des textes encodés en XML-TEI, disponible à l'adresse suivante : *TEI Publisher*, URL : `https://teipublisher.com/index.html` (visité le 14/08/2022).

l'importance historique de cette période : le mouvement populiste du boulangisme a fait trembler la République, diverses actions ont eu lieu telles que le ralliement des catholiques à la République, la montée du socialisme et du syndicalisme, et les premiers attentats anarchistes.

Une fois le traitement de ce sous-corpus effectué, et la preuve de concept confirmée, l'objectif d'AGODA sera d'appliquer la chaîne de traitement sur les autres données du corpus initial. Celui-ci pourra également être agrandi, en envisageant un projet de plus grande envergure prenant en compte l'ensemble des transcriptions des débats parlementaires tenus au Sénat durant la III^{e} République, ainsi que les Lois et décrets, eux aussi publiés dans le *Journal officiel*.

Chapitre 3

Océriser et explorer les débats, deux étapes effectuées avant l'encodage

3.1 Océriser les débats parlementaires

La première étape du projet consiste à océriser les débats parlementaires numérisés. Il s'agit d'extraire le texte de ces images via la reconnaissance optique de caractères. Cette étape est primordiale avant de pouvoir effectuer l'encodage des textes. Nous allons voir en quoi elle consiste et quels ont été les difficultés rencontrées et les choix effectués.

3.1.1 La reconnaissance optique de caractères

Le projet AGODA utilise comme source les débats parlementaires numérisés mis en ligne sur Gallica. Comme nous l'avons vu précédemment, cette bibliothèque numérique met à disposition de l'utilisateur une numérisation en mode image. Cette dernière consiste à scanner chaque page, et rend pour chacune d'elle une « photographie », ou plus précisément un fichier numérique en mode image. Gallica propose également une numérisation en mode texte. Cette dernière est le résultat de la conversion de l'image du texte numérisé en un format de texte lisible par l'ordinateur. Ce processus est effectué grâce à la reconnaissance optique de caractères (ROC ou OCR pour *Optical Character Recognition*). Selon Ahmed Ben Salah :

> Les systèmes de reconnaissance de caractères (OCR) sont des logiciels fondés sur des algorithmes permettant de passer d'un signal contenant des informations textuelles (images de caractères) à une forme de texte électronique codé au format ASCII, UTF-8, Unicode ou structurée au format XML. [1]

1. Ben Salah (Ahmed), *Maîtrise de La Qualité Des Transcriptions Numériques Dans Les Projets de Numérisation de Masse*, Theses, Université de Rouen, 2014, URL : `https://hal-bnf.archives-ouvertes.fr/tel-01164698` (visité le 11/08/2022).

Pour effectuer cette conversion automatique de l'image vers le texte, la reconnaissance optique de caractères repose sur une chaîne de traitement constitué de plusieurs opérations successives. Elle commence généralement par une étape de pré-traitement consistant à préparer les images contenant les textes afin d'en faciliter l'analyse ensuite par le logiciel d'OCR. Une phase de segmentation est ensuite réalisée pour déterminer les zones graphiques (tableau, illustration, etc.) et textuelles de l'image (page, lignes, mots, caractères). Le résultat de la segmentation des mots est ensuite envoyé au moteur de reconnaissance optique de caractères afin qu'il détermine l'identité des caractères. Enfin, une phase de post-correction est effectuée par ce dernier afin de valider les résultats ou de les corriger en utilisant un dictionnaire ou un modèle de langage [2].

Le projet AGODA s'est intéressé à l'OCR, car ce processus s'applique sur des numérisations de documents typographiques, contrairement à l'*Handwritten Text Recognition* (HTR) qui se base sur la numérisation des documents manuscrits.

3.1.2 Une source complexe à océriser

L'objectif pour AGODA est d'obtenir un texte océrisé de qualité, le plus conforme possible au texte de l'image. La qualité du taux de reconnaissance des caractères dépend de 2 facteurs : la qualité de la saisie lors de la numérisation des documents et les précautions prises lors de la mise en œuvre de l'océrisation [3]. C'est avec ce premier point que l'équipe du projet a rencontré le plus de difficultés.

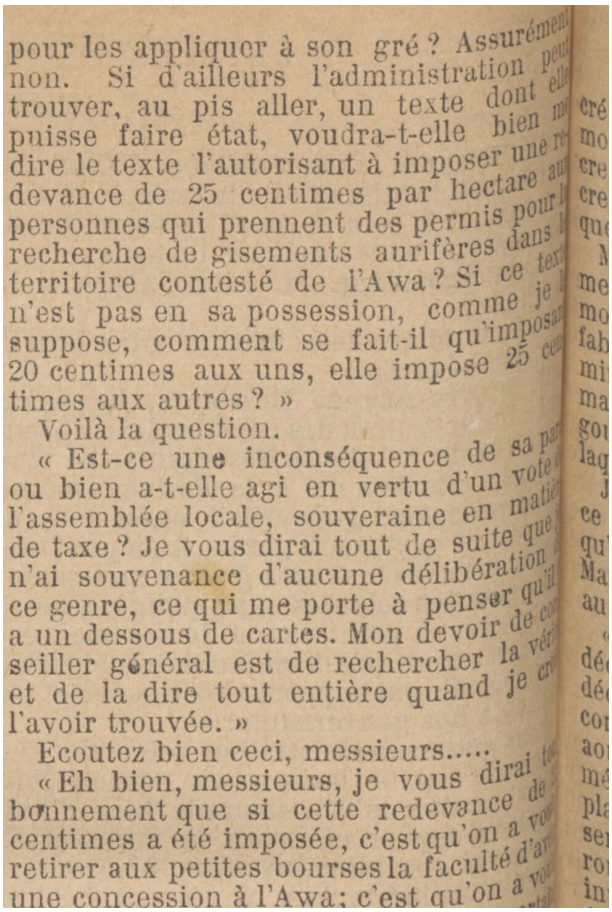
Dans un premier temps, AGODA a souhaité utiliser le résultat de l'océrisation effectuée par la Bibliothèque nationale de France. Ils ont récupéré, via l'API document de Gallica [4] les textes océrisés. Mais les membres de l'équipe se sont vite rendu compte que les taux de reconnaissance estimés à plus de 99 % par la BnF pour la plupart des textes se sont révélés être un indicateur trompeur qui dissimulait des problèmes de qualité de l'océrisation pour certaines pages ou certains numéros. La cause de ces erreurs était due à la qualité de la numérisation. En effet, cette dernière dépend de plusieurs facteurs déterminants comme la résolution, le contraste et la luminosité, l'inclinaison, mais aussi la qualité du support des documents [5]. Or, certains numéros du *Journal officiel* avaient des tâches ou des ombres. De plus, comme les numéros du journal étaient reliés sous forme de registres, les pages pouvaient être courbées, notamment celles situées au centre des registres. Ces particularités du support physique des documents, visibles sur les numérisations, ont donc eu un impact considérable sur la qualité de l'OCR.

2. *Ibid.*

3. Belaïd (Abdel), Pierron (Laurent), Najman (Laurent) et Reyren (Dominique), *La numérisation de documents : principes et évaluation des performances*, 2000, 35 p, URL : `https://hal.inria.fr/inria-00099148` (visité le 11/08/2022).

4. Disponible à l'adresse suivante : `https://api.bnf.fr/fr/api-document-de-gallica`.

5. Belaïd, *et al.*, *La numérisation de documents*, op. cit.



pour les appliquer à son gré ? Assuréme
non. Si d'ailleurs l'administration peu
trouver, au pis aller, un texte dont elle
puisse faire état, voudra-t-elle bien m
dire le texte l'autorisant à imposer une re
devance de 25 centimes par hectare au
personnes qui prennent des permis pour
recherche de gisements aurifères dans
territoire contesté de l'Awa ? Si ce tex
n'est pas en sa possession, comme je
suppose, comment se fait-il qu'imposa
20 centimes aux uns, elle impose 25 cen
times aux autres ? »
Voilà la question.
« Est-ce une inconséquence de sa pa
ou bien a-t-elle agi en vertu d'un vote
l'assemblée locale, souveraine en matiè
de taxe ? Je vous dirai tout de suite que
n'ai souvenance d'aucune délibération
ce genre, ce qui me porte à penser qu'il
a un dessous de cartes. Mon devoir de co
seiller général est de rechercher la vér
et de la dire tout entière quand je cr
l'avoir trouvée. »
Ecoutez bien ceci, messieurs.....
« Eh bien, messieurs, je vous dirai to
bonnement que si cette redevance de
centimes a été imposée, c'est qu'on a vo
retirer aux petites bourses la faculté d'a
une concession à l'Awa ; c'est qu'on a vo

(a) Numérisation d'une page courbée

pour les appliquer à son gré ? Assure.

non. Si d'ailleurs l'administration fj, trouver, au pis aller, un texte (1011t eu puisse faire état, voudra-t-elle dire le texte l'autorisant à imposer u devance de 25 centimes par hectar , personnes qui prennent des permis Pour1 recherche de gisements aurifères a teSI territoire contesté de l'Awa ? Si ce tes

n'est pas en sa possession, cornas le suppose, comment se fait-il qulIWjs3j 20 centimes aux uns, elle impose 20 cet times aux autres ? » , Voilà la question. é

« Est-ce une inconséquence de sa te ou bien a-t-elle agi en vertu d'un v l'assemblée locale, souveraine en OlqallC de taxe? Je vous dirai tout de suite Il n'ai souvenance d'aucune délibératl°ll'V ce genre, ce qui me porte à penser a un dessous de cartes. Mon devoir seiller général est de rechercher la cJ'!" et de la dire tout entière quand Je l'avoir trouvée. »

Ecoutez bien ceci, messieurs. l «Eh bien, messieurs, je vous dIra de bonnement que si cette redevance J centimes a été imposée, c'est qu'on a, retirer aux petites bourses la faculté" une concession à l'Awa; c'est qu'on a r¡;:i\

(b) Océrisation correspondante mettant en évidence les conséquences des courbures

FIGURE 3.1 – Extrait du compte rendu *in extenso* de la séance du 20 octobre 1890

Plusieurs solutions ont été mises en place afin de pallier ces difficultés pour obtenir un texte océrisé de qualité. L'objectif premier a consisté à améliorer la qualité de l'image avec une méthode de *dewarping*, cette dernière permettant le redressement des images, et donc la correction des courbures. Les premiers essais n'ont pas donné de résultat probant, c'est pourquoi l'équipe s'est tournée vers l'outil de nettoyage créé par le LRDE. Celui-ci permettait de mieux gérer le redressement des courbures, mais aussi de faire disparaître les taches présentes. Cette amélioration de l'image a permis de réocériser les textes dans de meilleures conditions et a favorisé en cela l'amélioration de la qualité de l'OCR. L'étape de réocérisation des textes a reposé sur plusieurs essais d'océrisation, à l'aide de deux moteurs d'OCR *open source*, la version proposée par HumaNum d'Abby FineReader et

Tesseract, afin de comparer leurs résultats, et de choisir le plus performant. Ces deux moteurs d'OCR ont permis d'obtenir des textes océrisés de meilleure qualité, mais ils n'ont pas été retenus pour la suite du projet [6].

Enfin, l'équipe du projet avait conscience que même si elle trouvait un OCR très performant, elle n'obtiendrait pas un texte parfaitement océrisé. Elle a donc réfléchi, en parallèle des océrisations, à une phase de post-correction afin de corriger automatiquement l'ensemble des textes obtenus [7]. Par cette phase, l'équipe aimerait obtenir des textes avec très peu d'erreurs afin de garantir à l'utilisateur de la plateforme future une consultation de textes contenant un nombre d'erreurs aussi faible que possible à l'issue de la post-correction.

3.1.3 Le choix de l'outil d'OCR du LRDE

Présentation de l'outil

Le projet AGODA s'est tourné vers l'outil d'OCR développé par Joseph Chazalon et Edwin Carlinet, membres du LRDE. Cet outil a été conçu dans le cadre de l'ANR SoDUCo [8], et paramétré pour océriser des annuaires et bottins, publiés au XIX^{e} siècle et dans la première moitié du XX^{e} siècle. Il utilise le moteur PERO OCR, qui est particulièrement performant sur les textes imprimés historiques. Cet outil est actuellement disponible en version alpha privée [9] et est amélioré quotidiennement par ses concepteurs en fonction des besoins des utilisateurs.

Justification du choix

Plusieurs raisons ont poussé l'équipe à choisir cet outil. Tout d'abord, il permettait d'obtenir des résultats d'océrisation très corrects sur les documents du projet. En effet, comme l'OCR a été entraîné sur des annuaires du XIX^{e} siècle, et que ces derniers avaient des similarités structurelles et formelles (colonnes, fonte de caractères Didot) avec les débats parlementaires, cela a eu un impact positif sur le résultat de l'OCR. De plus, l'outil d'OCR du LRDE offrait plusieurs fonctionnalités très utiles pour le projet. Il permettait à la fois d'océriser les textes, en identifiant d'abord la page, puis les blocs de textes (colonnes, paragraphes), puis les lignes, et en reconnaissant les entités nommées, comme les noms et rôles des locuteurs, les noms des lieux, les noms des organisations, et les nombres à

6. Nous expliquerons ce choix dans la sous-partie suivante 3.1.3.

7. Pour plus d'informations sur le détail des étapes de ré-océrisation, sur l'évaluation de la performance des deux systèmes d'OCR, et sur les méthodes de post-correction, se référer à l'article suivant : Puren, *et al.*, « Between History and Natural Language Processing », op. cit.

8. Le projet ANR SoDUCo, ou Dynamiques Sociales en contexte urbain : outils, modèles et données libres – Paris et ses banlieues, 1789-1950, a pour but d'étudier l'évolution de Paris dans le temps, aussi bien sur ses transformations morphologiques que sur ses évolutions sociales.

9. Puren, *et al.*, « Between History and Natural Language Processing », op. cit.

l'aide du modèle linguistique CamemBERT [10]. Cette reconnaissance des entités nommées est un élément particulièrement utile pour le projet, car nous souhaitons extraire ces informations. De même, l'outil d'OCR peut reconnaître certaines marques de structure comme les titres. En outre, en plus de fournir ces fonctionnalités précieuses, l'outil du LRDE propose une interface claire et facile d'utilisation pour le traitement des textes.

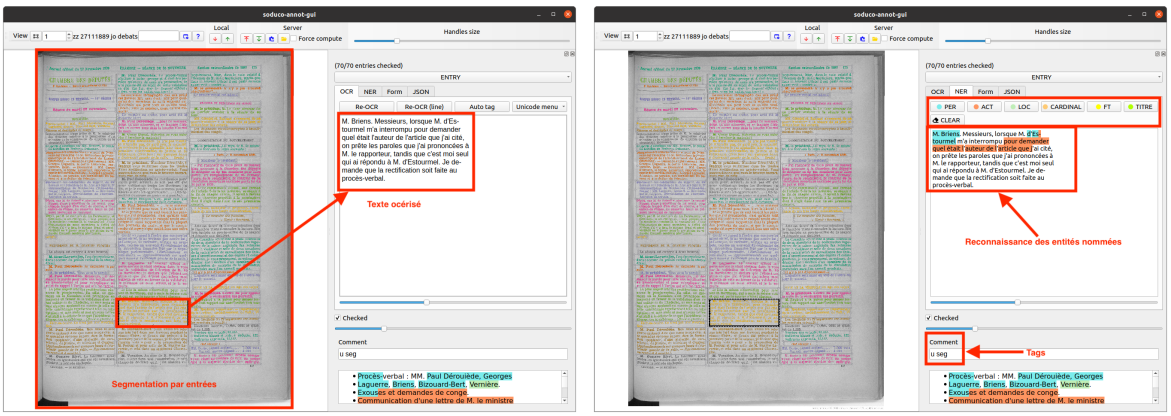


FIGURE 3.2 – Fonctionnalités de l'interface de l'outil d'OCR du LRDE

Après avoir téléchargé les images numérisées dans l'interface, il est possible de visualiser séparément ou simultanément l'image, selon ces différents niveaux de segmentation, et le texte océrisé. L'interface permet également de corriger manuellement la segmentation par un recadrage des différents éléments structurels identifiés (titres, paragraphes, titre courant, etc.) - aussi appelés « *boxes* » (pour « boîtes » en anglais) dans le cadre du développement du logiciel, de relancer l'océrisation en fonction des corrections apportées, de corriger et annoter avec des étiquettes ou « tags » les textes océrisés, et d'exporter le résultat sous le format JSON. Enfin, l'équipe du projet AGODA a favorisé l'utilisation de cet outil pour des raisons pratiques. En effet, le LRDE était situé dans les mêmes locaux que le laboratoire MNSHS. Il était alors possible de se réunir et d'échanger régulièrement avec les deux concepteurs de l'outil.

3.2 Explorer les débats parlementaires

Le projet AGODA a pour objectif d'améliorer la connaissance des débats parlementaires. Au-delà des missions liées à l'accessibilité de la source, il place au cœur de ses intérêts l'analyse de cette dernière. Nous allons voir quelles ont été les méthodes d'analyses mises en place sur les textes océrisés et comment elles seront valorisées.

10. CamemBERT est un modèle de langue pré-entraîné sur un jeu de données francophone, permettant justement de reconnaître les entités nommées, mais aussi d'étiqueter les parties du discours, et d'analyser la syntaxe, `http://almanach.inria.fr/software_and_resources/default/CamemBERT-fr.html`.

3.2.1 Analyser les débats par le *topic modeling* et le *word embedding*

AGODA souhaite contribuer à l'exploration des débats parlementaires. Pour cela, l'équipe du projet a effectué des analyses à l'aide de deux méthodes computationnelles : le *topic modeling* et les *words embeddings*[11]. Ces deux méthodes, particulièrement appropriées pour l'analyse de grand corpus, ont été appliquées à partir des textes océrisés fournis par la BnF (sans post-correction) et sur un corpus plus large (1881-1899) que celui défini au préalable, car elles nécessitent une grande quantité de texte.

Tout d'abord, le *topic modeling*, ou modélisation de sujets, a été perçu comme un « point d'entrée » intéressant dans les débats parlementaires. Cette méthode d'apprentissage non supervisée permet de découvrir des sujets abstraits, dits « topics », issus d'un corpus à l'aide d'un modèle probabiliste. L'analyse qui a été effectuée est basée sur le modèle probabiliste bayésien LDA (*Latent Dirichlet Allocation*). Ce dernier repose sur l'hypothèse théorique suivante : les sujets correspondent à des champs sémantiques, autrement dit à des ensembles de mots liés par leur signification. Ces sujets préexistent avant l'écriture d'un texte. Les textes sont ensuite construits à l'aide de mots issus de sous-ensembles de sujets, et son auteur ne fait que puiser dans ces sujets pour écrire son texte. Par exemple, un texte portant sur le transport maritime va puiser dans les champs sémantiques ou « sujets » ayant trait à la mer, aux navires et plus largement aux moyens de transports. Le rôle de LDA est d'inverser le processus de génération, en partant du texte pour extraire les sujets. L'algorithme a permis d'obtenir des listes de mots, chaque liste correspondant à un sujet. Le nom de ces listes a ensuite été défini manuellement en fonction du vocabulaire de chaque liste[12]. Ainsi, la modélisation de sujets, en extrayant les sujets du corpus, peut être un bon moyen pour analyser l'évolution des différents thèmes évoqués dans les débats parlementaires, et permettre de mieux comprendre l'évolution des idées politiques dans le temps. C'est ce que l'équipe du projet a tenté de démontrer dans l'article *Using Topic Generation Model to explore the French Parliamentary Debates during the early Third Republic (1881-1899)*[13].

Par ailleurs, le *word embedding*, ou plongement de mots, est une méthode d'apprentissage de représentation de mots. Elle a été réalisée par l'équipe afin d'améliorer et de compléter l'analyse effectuée sur les débats parlementaires par la modélisation de sujets.

11. Ces deux méthodes computationnelles s'inscrivent dans le domaine du Traitement Automatique des Langues (TAL).

12. Bourgeois (Nicolas), Pellet (Aurélien) et Puren (Marie), « Using Topic Generation Model to Explore the French Parliamentary Debates during the Early Third Republic (1881-1899) », dans *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)*, dir. Matti La Mela, Fredrik Norén et Eero Hyvönen, 2022 (CEUR Workshop Proceedings), p. 35-51, URL : `https://hal.archives-ouvertes.fr/hal-03526254` (visité le 09/08/2022).

13. *Ibid.*

Le *word embedding* permet de représenter des mots dans un espace vectoriel en fonction de leur similarité sémantique. Cette méthode repose, en effet, sur la théorie linguistique de la *Distributional Semantics*. Cette dernière considère qu'un mot est caractérisé par son contexte, autrement dit par les mots qui l'entourent. Cela signifie que si les mots partagent des contextes similaires, ils partagent alors aussi des significations similaires. Ainsi, plus un mot est proche d'un autre mot dans l'espace vectoriel, plus ils sont proches sémantiquement [14]. Plusieurs approches et algorithmes de *word embedding* existent. L'équipe du projet en a utilisé plusieurs et les détaille dans l'article : *Between History and Natural Language Processing : Study, Enrichment and Online Publication of French Parliamentary Debates of the Early Third Republic (1881-1899)* [15].

3.2.2 Valorisation des analyses

Ces deux méthodes computationnelles ont permis à l'équipe du projet d'effectuer de nouvelles analyses sur les débats parlementaires, en approchant le corpus des textes dans son ensemble. Ces résultats pourront ensuite être valorisés de différentes façons auprès des chercheurs et du grand public.

Une première approche consiste à mettre en évidence des sujets sur la future plateforme de consultation. En effet, elles pourront être intégrées dans l'encodage en XML-TEI des textes, afin d'apporter ensuite de nouvelles fonctionnalités de recherche thématiques. L'utilisateur pourra par exemple se référer à la liste des sujets afin de suivre dans le temps l'évolution des débats sur une thématique particulière ou d'étudier l'évolution des positions d'un parlementaire ou d'un groupe politique sur une thématique. Ces sujets pourront être visualisés aussi, au sein des textes, grâce à la surbrillance des termes associés à ces derniers. Cet objectif permettra alors de répondre à l'un des objectifs fixé par AGODA : la création de sous-corpus à l'aide des sujets. Il contribuera aussi à la mise en place d'une nouvelle approche des textes, s'inscrivant dans le concept de « *distant reading* » (lecture distante) de Franco Moretti [16]. Ce concept permet, en effet, d'interpréter les textes selon une lecture différente, en les parcourant de façon « distante » avec des outils numériques. À l'aide des sujets, la future plateforme de consultation des débats pourra alors permettre un aller-retour entre lecture proche et lecture distante, et contribuer en cela à l'essor de nouvelles perspectives de recherche [17].

Une autre forme de valorisation reposerait sur des outils de visualisation des résultats des analyses réalisées (tableau, graphique, figure). Mais cette question reste encore à discuter ; car par nature, TEI Publisher est conçue pour publier des textes et pas des résultats de recherche.

14. Définition issue du site web suivant : `https://dataanalyticspost.com/Lexique/word-embedding/`.

15. Puren, *et al.*, « Between History and Natural Language Processing », op. cit.

16. Concept développé dans : Moretti (Franco), *Distant Reading*, London ; New York, 2013.

17. Clavert (Frédéric), « Vers de nouveaux modes de lecture des sources », dans Olivier Le Deuff, *Le temps des humanités digitales : la mutation des sciences humaines et sociales*, fyp, 2017, p. 33-47.

Deuxième partie

Vers une structuration et un enrichissement des débats parlementaires : l'élaboration de l'encodage

La deuxième étape de la chaîne de traitement du projet AGODA consiste à encoder les comptes rendus des débats parlementaires de la Chambre des députés (1889-1893) [18] afin de structurer et d'enrichir sémantiquement les données. L'action d'encoder un texte signifie marquer avec des balises le contenu textuel de ce dernier : les balises apportent alors une indication explicite particulière sur l'élément balisé. Afin de pouvoir encoder le texte, autrement dit afin de pouvoir appliquer un ensemble de balises sur un texte, il est nécessaire d'élaborer les principes de l'encodage en amont. Nous avons réalisé cette élaboration en deux temps : nous avons d'abord mis en place une phase de modélisation afin de réfléchir à la forme que devait prendre l'encodage ; puis, dans un second temps, nous avons formalisé les choix réalisés en créant un schéma d'encodage et une documentation. En concertation avec les deux coordinateurs du projet, j'ai été chargée de réaliser ces deux missions lors de mon stage.

18. Les élections des députés de la V^{e} législature ont eu lieu les 22 septembre et 6 octobre 1889. La législature a débuté le 12 novembre et s'est terminée le 14 octobre 1893. L'encodage a donc été pensé pour les comptes rendus propres à cette période.

Chapitre 4

Modéliser l'encodage

La modélisation est une étape primordiale dans le processus d'élaboration d'un encodage. Elle correspond à une phase de réflexion et d'essai, permettant d'établir un modèle selon les différents objectifs visés. Elle constitue en cela un travail préparatoire qui aide à l'élaboration de l'encodage : « Les modèles et la modélisation constituent des aides à l'élaboration et à la structuration des idées ; ce sont des supports au raisonnement. [1] ». Nous avons divisé cette phase de modélisation en plusieurs étapes avant de créer le modèle lui-même : nous avons d'abord fixé un cadre scientifique et technique à l'encodage, puis nous avons fait un travail d'analyse important afin d'évaluer les différentes possibilités de balisage par rapport à notre source. Cette évaluation a été poursuivie enfin lors de la création d'un encodage test.

4.1 Donner un cadre

La réalisation d'un encodage nécessite de définir au préalable des objectifs scientifiques et techniques. L'équipe d'AGODA s'est alors questionnée sur le sens et le but de l'encodage qu'elle voulait créer, et sur les possibilités techniques et les moyens pratiques pour le mettre en place.

4.1.1 Fixer des objectifs

Dans un premier temps, les membres de l'équipe ont déterminé les informations majeures qu'ils souhaitaient encoder, et par conséquent, les informations qu'ils voulaient exploiter. Ces dernières étaient constituées :

— des entités nommées, avec le référencement des personnes, des lieux et des institutions ;

— des interventions des députés et les interventions des groupes parlementaires, grâce à l'encodage des prises de paroles ;

1. Calderan, Hidoine et Millet, *Métadonnées*, op. cit.

— de la structure, avec la mise en évidence des parties et titres structurant le texte ;
— des annexes, en les distinguant du corps du texte et en les structurant.

Ces informations ont également été complétées par d'autres enjeux d'exploitabilité à intégrer dans l'encodage, à savoir les sujets, à l'aide des analyses de *topic modelling* et de *word embeddings*, les segments de texte liés aux mouvements, partis et idées politiques, et les annotations linguistiques. Par l'encodage de tous ces éléments, le projet AGODA avait pour but de structurer et d'enrichir sémantiquement les données, afin de pouvoir extraire plus facilement et d'analyser les informations souhaitées.

Ces différents objectifs scientifiques donnent un cadre à l'encodage. Celui-ci devait être pensé en fonction de ces derniers. Toutefois, la réalisation de ces objectifs dépendait de deux paramètres à prendre en compte :

— le format de l'encodage (validité selon le format choisi) ;
— les moyens à disposition pour le réaliser (temps, main-d'œuvre, méthode).

Les choix d'encodage, contraints par ces deux paramètres, devaient donc être pensés aussi en fonction de ce cadre technique et pratique. Les objectifs scientifiques définis au début du projet constituaient alors un but à atteindre, pouvant être révisés en fonction des autres objectifs définis.

4.1.2 Le choix du XML-TEI

Le projet AGODA avait pour objectif initial d'encoder le corpus à l'aide du format de balisage XML-TEI. Cette exigence a donc fourni également un cadre pour la modélisation de l'encodage. Nous allons justifier le choix de ce format, et exposer les contraintes de ce dernier pour notre encodage.

eXtensible Markup Language

L'*Extensible Markup Language* (XML), ou « langage de balisage extensible » en français, est un métalangage de balisage générique permettant de structurer les données. Créé par le *World Wide Web Consortium* (W3C) en 1998 et issu du langage de description à balises *Standard Generalized Markup Language* (SGML), le XML permet de favoriser la lisibilité des données aussi bien par les machines que par les humains. Il facilite aussi l'échange des données, et la migration vers d'autres plates-formes, logiciels ou formats [2]. Le XML a plusieurs principes fondamentaux. D'abord, il est construit à l'aide de balises marquées par des chevrons, ces dernières permettant de « caractériser […] le rôle intellectuel des mots, des groupes de mots, des phrases ou des portions de textes à l'intérieur de

2. Albouy (Ségolène), *Introduction à XML - Cours*, 13 oct. 2021, URL : `https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/3f4cb64280c009553e0db6c820ae228893699695/01-Introduction_XML/2021-10-13-Introduction_XML.pdf` (visité le 09/09/2022).

l'information [3] ». En effet, ces balises permettent de pointer « des parties spécifiques du flux de données pour indiquer une fonction structurelle ou des éléments de sémantique [4] ». La partie textuelle encodée avec une balise ouvrante et une balise fermante correspond à un élément XML [5], et ce dernier peut être caractérisé à l'aide d'un attribut [6]. De plus, le XML permet de hiérarchiser les éléments sous forme d'un arbre, un élément pouvant contenir, zéro, un ou plusieurs autres éléments, sans jamais se chevaucher. En outre, l'arbre XML permet aux éléments enfants d'hériter des propriétés des éléments parents. Ces particularités du XML peuvent être codifiées selon des spécifications. Ces dernières représentent en cela la « grammaire » du format XML ; elles donnent un modèle d'agencement des balises. Par ailleurs, le format XML ne précise pas le nom des balises qui le constitue. Comme le soutien Lou Burnard : il « ne dit rien sur la façon dont les éléments ou attributs doivent être nommés [...] et moins encore sur ce que leur nom signifie. [7] ». XML est une syntaxe, mais pas un vocabulaire. Il doit donc être associé à un (ou des) espace(s) de noms [8] permettant de nommer les éléments et les attributs.

Text Encoding Initiative

La *Text Encoding Initiative* (TEI) repose sur le langage XML [9] et est un « set de balises prédéfini et documenté [...] qui permet de procéder à une description « scientifique » et « sémantique » d'un texte [10] ». Elle fournit, en effet, le nom de nombreuses balises et des règles pour leur utilisation. La TEI est issue des travaux de chercheurs du Vassar College (États-Unis). Ces derniers se sont réunis en 1987 pour discuter de la faisabilité d'un schéma d'encodage standard [11] dans le but de répondre aux besoins de « structuration, de conceptualisation et de mise en réseau des textes dans le milieu de la recherche [12] ». Initiée et définie avec les principes « *Poughkeepsie* », la TEI a évolué et a pris une place importante au sein des humanités numériques. Elle en est actuellement à sa cinquième version (TEI P5).

3. Poupeau (Gautier), « L'édition électronique change tout et rien. Dépasser les promesses de l'édition électronique », *Le médiéviste et l'ordinateur* (, 18 avr. 2004), URL : `https://archivesic.ccsd.cnrs.fr/sic_00137222` (visité le 10/08/2022).

4. Burnard (Lou), *Qu'est-ce que la Text Encoding Initiative ?*, Marseille, 2015 (Encyclopédie numérique), URL : `http://books.openedition.org/oep/1237` (visité le 10/08/2022).

5. Nous utiliserons la graphie suivante afin d'indiquer le nom des éléments : <nomBalise>.

6. Nous utiliserons la graphie suivante afin d'indiquer le nom des attributs : @nomAttribut.

7. Burnard, *Qu'est-ce que la Text Encoding Initiative ?*, op. cit.

8. La particularité du XML est qu'il est un langage « extensible », autrement dit, une instance XML peut contenir un ou plusieurs espaces de noms.

9. Le langage TEI d'origine (P1 à P3) utilisait la syntaxe SGML.

10. Albouy (Ségolène), *Décrire Les Documents Patrimoniaux - Cours*, 25 oct. 2021, URL : `https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/03-XML_TEI/2021-10-25-TEI.pdf` (visité le 14/08/2022).

11. Se référer au chapitre « About These Guidelines » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, URL : `https://tei-c.org/` (visité le 14/08/2022).

12. Calderan, Hidoine et Millet, *Métadonnées*, op. cit.

La TEI définit un très grand ensemble d'éléments et d'attributs XML, afin de baliser toutes sortes de texte, peu importe la forme, la date, et la langue de ces derniers. Cette « encyclopédie de notions textuelles [13] » donne le nom et le cadre d'utilisation des balises, ces informations étant documentées au sein des *Guidelines*. Elle intègre deux catégories d'éléments : ceux spécifiques aux métadonnées du texte encodé (auteur et responsabilité, informations bibliographiques, description du manuscrit, historique des révisions, etc.), et ceux du texte lui-même (sections, titres, paragraphes, citations, etc.) [14]. L'ensemble des concepts qu'elle intègre est organisé en plusieurs groupes. Les modules sont des ensembles cohérents de balises permettant de regrouper d'une part les balises obligatoires et les plus courantes, et d'autre part, des balises spécifiques au type de texte encodé (poésie, théâtre, etc.) ou au type de données et d'informations (apparat critique, éléments graphiques, etc.). Par exemple, le module « drama » rassemble un ensemble de balises permettant d'encoder des pièces de théâtre ou des scripts radiophoniques. De plus, les classes permettent d'organiser les éléments du modèle, ces derniers héritant des propriétés de la classe à laquelle ils appartiennent. Il existe deux types de classes, celle permettant de regrouper les éléments qui partagent les mêmes attributs, et celle rassemblant les éléments qui sont situés aux mêmes endroits dans le document. La sous-classe « model.divPar » permet, par exemple, de regrouper les éléments pouvant être contenu par l'élément . En outre, les macros correspondent à des raccourcis qui spécifient des modèles de contenus les plus fréquents [15].

Les avantages du XML-TEI au regard des besoins d'AGODA

Les caractéristiques du format XML et les principes de conception de la TEI présentent de nombreux avantages qui répondent à nos enjeux et besoins d'encodage. C'est pourquoi le projet AGODA a choisi d'utiliser le XML-TEI. Tout d'abord, la TEI a été pensée par et pour les chercheurs et est destinée au domaine des sciences humaines et sociales. Selon Gautier Poupeau, elle permet :

> de disposer d'un schéma adapté à l'édition en sciences humaines et sociales, de proposer des balises correspondant à plus de 90 % des besoins d'un chercheur et d'un éditeur scientifique pour l'édition de sources, d'obtenir l'aide d'une communauté bien établie et d'utiliser un standard maintenu et mis à jour régulièrement. [16]

Elle propose, en effet, un ensemble de balises répondant à des besoins très diversifiés et pouvant s'appliquer sur tout type de texte. Elle s'engage à être compréhensible, flexible et

13. Burnard, *Qu'est-ce que la Text Encoding Initiative ?*, op. cit.

14. Se référer au chapitre « About These Guidelines » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, op. cit.

15. Albouy (Ségolène), *Les TEI Guidelines - Cours*, 22 nov. 2021, URL : `https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/e7c2aabedd885b98fbf11faacea32ff707eb01af/05-Les_TEI_Guidelines/2021-11-22-Guidelines.pdf` (visité le 01/09/2022).

16. Poupeau, « L'édition électronique change tout et rien. Dépasser les promesses de l'édition électronique », op. cit.

extensible [17]. Elle est aussi très régulièrement mise à jour par la communauté scientifique TEI. La TEI représentait donc une solution idéale pour le projet AGODA, puisqu'elle peut s'adapter aux caractéristiques des débats parlementaires, tout en répondant aux objectifs scientifiques des chercheurs et, car elle constitue un langage normalisé et pérenne, grâce à sa maintenabilité régulière [18].

Le XML-TEI permet aussi de garantir l'échange des données, un des enjeux souhaités par AGODA. De fait, le XML-TEI est un standard largement partagé dans le monde des humanités numériques, mais il est aussi un format ouvert et libre, ce qui lui permet d'être indépendant de tout environnement logiciel [19]. Le XML-TEI est donc fait pour être interopérable [20], dans le but de faciliter l'échange et la collaboration.

De surcroît, le XML-TEI propose différents niveaux d'encodage : visuel pour les aspects physiques et paléographiques des sources, sémantique pour le contenu intellectuel, et analytique pour la correspondance de certaines parties du texte avec une grille d'analyse spécifique (taxonomie, thesaurus, etc.). Par la même, il permet d'encoder les textes pour des usages multiples. Il peut servir à analyser les textes du point de vue de langue ou du style, ou encore lier le texte à une image, modifier, afficher et relier des documents grâce à des systèmes hypertextes, etc. [21]. Le fait de pouvoir manipuler les données dans des contextes variés a, là encore, justifié le choix du XML-TEI.

En outre, la TEI fournit des conseils pour l'encodage des textes au sein d'une documentation rédigée. Cette dernière permet d'aider les encodeurs dans leur démarche, en leur fournissant des définitions et des conseils d'utilisation selon le contexte, tout en essayant de tenir compte de points de vue très différents. La richesse des *Guidelines* représentait ainsi un atout pour AGODA, permettant de faciliter la modélisation de son encodage.

Un cadre pour l'encodage

Construire un encodage selon un format particulier nécessite de prendre en compte les règles qui le définissent. Le XML-TEI constitue donc en cela un cadre technique et réflexif, l'objectif étant que l'encodage soit « conforme » et « valide » par rapport à celui-

17. Se référer au chapitre « About These Guidelines » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, op. cit.

18. Selon G. Poupeau : « L'utilisation d'un standard n'est pas un gage de sa pérennité à très long terme, mais un standard étant maintenu et normalisé, on peut espérer la mise en place d'outils de conversion qui permettront de migrer les données dans les futurs standards informatiques. ». La maintenabilité de la TEI par la communauté favorise donc sa pérennité à long terme. Poupeau, « L'édition électronique change tout et rien. Dépasser les promesses de l'édition électronique », op. cit.

19. Albouy, « Décrire Les Documents Patrimoniaux - Cours », op. cit.

20. L'interopérabilité est « un état qui existe entre deux applications quand, pour une tâche spécifique, une application peut accepter les données d'une autre application pour effectuer cette tâche, de manière appropriée et satisfaisante, sans que cela nécessite l'intervention d'un opérateur extérieur. ». Définition issue de : Calderan, Hidoine et Millet, *Métadonnées*, op. cit.

21. Se référer au chapitre « About These Guidelines » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, op. cit.

ci. Pour qu'un encodage soit dit « conforme », il faut qu'il respecte les règles syntaxiques inhérentes au XML. Nous pouvons citer les quatre règles principales suivantes :

— une balise ouvrante doit toujours être close par une balise fermante ;
— aucun enchevêtrement n'est autorisé entre les éléments ;
— le document XML doit commencer par une déclaration et par un seul élément racine, précisant l'espace de nom ;
— un élément ne peut pas contenir deux attributs du même nom.

Dans un second temps, un encodage peut être aussi dit « valide », s'il se conforme à un schéma, ou à un dictionnaire de balises. Dans notre cas, puisque la TEI ne constitue pas un schéma à proprement parlé, mais plutôt un *framework* permettant de construire son propre schéma [22], il s'agit d'abord d'être conforme à la TEI, en respectant le modèle abstrait qu'elle institue, c'est-à-dire la fonction sémantique attribuée aux éléments [23] et leur contexte d'utilisation, et respecter bien évidemment la syntaxe XML. Ensuite il est possible d'obtenir un encodage « valide » par rapport aux spécifications TEI définies dans un schéma particulier qui lui est associé.

Pour illustrer ces propos, nous pouvons citer deux règles définissant un document conforme au regard du XML-TEI :

— le document doit commencer par une déclaration XML et l'élément racine TEI, ce dernier précisant l'espace de nom TEI ;
— le document doit ensuite contenir au moins deux parties : l'en-tête contenant des métadonnées décrivant le document, et représenté à l'aide de l'élément <teiHeader>, et le texte lui-même, contenu dans l'élément <text>.

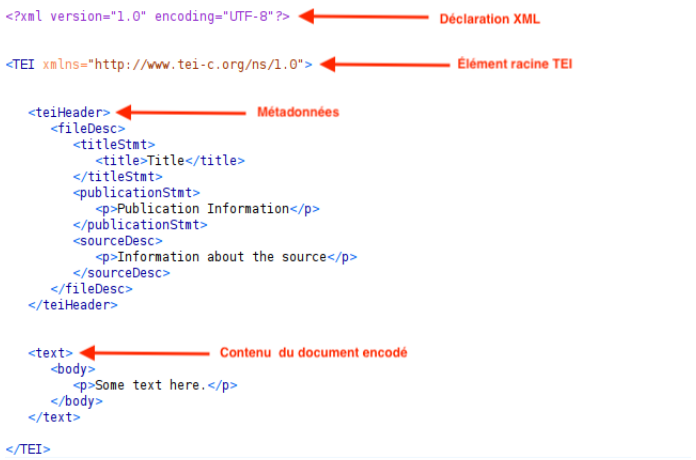


FIGURE 4.1 – Structure initiale d'un document XML-TEI

22. Albouy, « Décrire Les Documents Patrimoniaux - Cours », op. cit.
23. Se référer au chapitre « Using the TEI » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, op. cit.

4.1.3 Les moyens pratiques

L'équipe du projet AGODA a également dû prendre en compte, dans le cadre de la modélisation de l'encodage, les moyens à sa disposition pour le réaliser. Si l'objectif majeur est en effet de pouvoir parvenir à un encodage, sa conception et sa réalisation dépendent de plusieurs facteurs pratiques. Le temps a été la première condition à prendre en considération. Le projet AGODA est financé pour 1 an et dispose donc d'un temps limité pour réaliser ses missions. Par conséquent, l'encodage doit être réalisable sur le temps imparti, ce qui a impacté notre réflexion sur le degré de précision d'encodage à établir.

Par ailleurs, il a été nécessaire de tenir compte de la main-d'œuvre à disposition, puisque la réalisation de l'encodage dépend d'elle, et de la complexité de la source sur le plan quantitatif. Au vu du nombre de personnes dans l'équipe du projet, mais aussi et surtout au vu du nombre de textes à encoder, le projet AGODA s'est tourné vers l'automatisation de l'encodage. Cette méthode, complexe à mettre en place, nous a donc fait réfléchir là encore sur la conception de l'encodage, et principalement sur le degré de précision à modéliser.

Malgré ces deux facteurs, nous avons souhaité garder, tout de même, un degré de précision d'encodage important. Nous avons pris le parti de modéliser un encodage « idéal », qui devait être précis et prendre en compte un maximum des objectifs scientifiques établis. Mais nous savions qu'il y aurait un écart entre cette modélisation de l'encodage et sa réalisation, puisque nous avions bien conscience que son application serait complexe et qu'elle constituerait un défi vers lequel tendre le plus possible.

4.2 Analyser les possibilités

Le travail de modélisation a consisté dans un deuxième temps à l'analyse des possibilités d'encodage des débats parlementaires. En effet, les choix d'encodage nécessitent en amont un long processus d'analyse, que nous avons axé d'abord sur la connaissance approfondie des particularités textuelles de la source, puis sur les différentes possibilités offertes par le standard d'encodage choisi pour répondre à nos besoins spécifiques. Pour cela, nous avons pris en compte d'une part les projets d'encodage réalisés en XML-TEI sur des sources similaires, et d'autre part nous avons complété ces analyses par l'étude des *guidelines* de la TEI.

4.2.1 Analyser les particularités du corpus

Les comptes rendus *in extenso* publiés dans le *Journal officiel* font partie d'une typologie particulière, définie en première partie. Cette typologie a des caractéristiques textuelles que nous avons dû analyser afin de pouvoir réaliser des choix d'encodage adaptés

à celles-ci. La consultation approfondie d'un ensemble conséquent de numéros issus du corpus [24] nous a permis de mettre en évidence les points suivants.

Chaque numéro du *Journal officiel* correspond à une séance parlementaire [25]. Le numéro est publié au lendemain de la séance. Il a une forme, une structure et un contenu bien définis qui restent, dans la majorité des cas, identiques. La source présente ainsi une construction plutôt homogène.

FIGURE 4.2 – Analyse de la première page du compte rendu de la séance parlementaire du 26 novembre 1889

24. L'analyse a été effectuée majoritairement à partir des numéros de la V^e législature, mais il n'y a pas de changements fondamentaux sur l'ensemble de la III^e République.

25. Certains premiers numéros de l'année peuvent contenir plusieurs séances, se référer au fichier FR_3R_5L_1890-01-14_digitisation.pdf en annexe A.1.

FIGURE 4.3 – Analyse d'un extrait des annexes du compte rendu de la séance parlementaire du 26 novembre 1889

Analyse formelle

D'un point de vue formel, une page contient un bandeau, le texte disposé sur trois colonnes et parfois un pied de page. La première page du numéro possède également une manchette.

Le bandeau de la première page est constitué du nom du journal suivi de la date de publication, du terme « Chambre » en majuscule complété par la date de la séance, du type de session parlementaire suivi de l'année de cette dernière et enfin du numéro de page. Le bandeau des pages suivantes est plus succinct : l'information « Chambre des

députés » est donnée en majuscule et est suivie de la date de la séance, le numéro de page est indiqué.

La manchette constitue le titre de la publication. Elle mentionne le nom de l'Assemblée, le numéro de législature, le type de session et sa date, l'expression « Compte-rendu *in extenso* », le numéro de séance[26] et enfin la date de la séance, précédée de l'expression « Séance du ». La manchette est généralement contenue sur le haut de la première colonne, mais son emplacement peut varier sur certains numéros, elle peut être située parfois au-dessus des colonnes.

Le texte disposé sur trois colonnes se divise en paragraphes, chacun commençant par un alinéa. Certaines parties du texte peuvent être mises en avant à l'aide d'éléments typographiques tels que les graisses (italique, gras) et les petites capitales. La signature du chef sténographique est indiquée avant le développement des parties complémentaires.

Le pied de page est présent sur certaines pages. Il donne des informations mineures liées spécifiquement à la publication du journal : il contient l'expression « Chambre. – In extenso », le numéro de feuillet, et sur la dernière page, il mentionne les informations sur l'impression.

Analyse structurelle

D'un point de vue structurel, le compte rendu commence toujours avec un sommaire, annonçant les différents sujets qui seront abordés. Il est suivi du corps du texte contenant les propos tenus lors de la séance. Il se conclut avec le règlement de l'ordre du jour et est complété par des parties complémentaires. Ces dernières peuvent être de différentes sortes : les annexes contenant le détail des scrutins (résultat des votes, nom des votants, leurs positions, etc.), les rapports, les erratum, les rectifications, etc. Le texte du compte rendu est divisé en parties, introduites régulièrement avec un titre en majuscule. Ces parties peuvent contenir, outre du texte, des informations données sous forme de tableau.

Analyse sémantique

D'un point de vue sémantique, et comme affirmé dans la partie précédente, le compte rendu des discussions comprend les interventions des députés, chaque intervention commençant par le nom et/ou la fonction de l'orateur en gras suivi(e) d'un point. Leurs propos pouvaient être ponctués d'incidents divers (applaudissements, bruits, exclamations, etc.). Ces derniers sont indiqués entre parenthèses et en italique. Des commentaires portant sur la description du fonctionnement de la Chambre des députés (ouverture et fermeture de la séance, début d'un vote, etc.) sont ajoutés en plus des interventions.

26. La numérotation des séances recommence au début de chaque session.

4.2.2 Analyser des projets similaires

En parallèle de ce travail, et afin de nous aider dans le choix des balises, nous nous sommes intéressés aux projets ParlaClarin et ParlaMint. Ces deux projets produisent des corpus de débats parlementaires contemporains multilingues, encodés en XML-TEI.

Présentation des projets

ParlaClarin est un projet de l'infrastructure de recherche CLARIN pour les ressources et technologies linguistiques. Il a été initié en réponse au manque de standardisation pour l'encodage des débats parlementaires. En effet, les corpus existants étaient encodés de multiples façons et selon des formats différents, ce qui posait un important problème pour la comparaison, l'échange et la réutilisation des données [27]. Face à ce constat, CLARIN a organisé en 2019 un atelier, « CLARIN ParlaFormat », au cours duquel les participants ont pu partager leur expérience en matière d'encodage de corpus parlementaires et discuter des solutions à adopter [28]. Le projet ParlaClarin a élaboré un schéma de structuration de données limitant les options d'encodage de la TEI pour des options applicables spécifiquement aux débats parlementaires. Ils ont également élaboré des recommandations textuelles afin d'expliciter les choix d'encodage possibles. Ils ont souhaité favoriser l'échange des documents en choisissant une approche « descriptive », et ont ainsi préféré construire un schéma souple et adaptable. Ils ont mis, en cela, l'enjeu de l'interopérabilité des données de côté [29]. ParlaClarin met à disposition le schéma XML, ses lignes directrices et des exemples de corpus encodés sur un dépôt GitHub [30].

Le projet ParlaMint s'est constitué en novembre 2019 suite à l'initiative de ParlaClarin. Il a développé jusqu'à présent 27 corpus de débats parlementaires contemporains en XML-TEI, portant sur la pandémie de COVID, avec 16 langues principales. ParlaMint propose une spécialisation des recommandations de ParlaClarin. Il adopte, a contrario, une approche plus « prescriptive » et donc restrictive. Il impose une structure rigide notamment sur les noms des fichiers, la structure des dossiers, et l'encodage lui-même. Il favorise donc, en plus de l'échange des données, leur interopérabilité [31]. ParlaMint met à

27. Pančur (Andrej) et Erjavec (Tomaž), *Parla-CLARINA TEI Schema for Corpora of Parliamentary Proceedings*, 3 mai 2022, URL : `https://clarin-eric.github.io/parla-clarin/` (visité le 14/08/2022).

28. Pour plus d'informations, se référer au site web suivant : `https://www.clarin.eu/blog/clarin-parlaformat-workshop`.

29. Pančur et Erjavec, *Parla-CLARINA TEI Schema for Corpora of Parliamentary Proceedings*, op. cit.

30. Clarin (Éric), *Parla-CLARIN*, GitHub, URL : `https://github.com/clarin-eric/parla-clarin` (visité le 14/08/2022).

31. Pančur (Andrej) et Erjavec (Tomaž), *The Structure and Encoding of ParlaMint Corpora*, 27 juill. 2022, URL : `https://clarin-eric.github.io/ParlaMint/` (visité le 30/08/2022).

disposition, comme ParlaClarin, le schéma XML, ses recommandations, et l'ensemble des corpus annotés sur un dépôt GitHub [32].

Le projet ParlaMint suit une grande partie des recommandations ParlaClarin, et peut donc être compatible avec le schéma de celui-ci. Il a pu également influencer certains choix de ParlaClarin. Ainsi, ces deux projets prennent en compte dans leur encodage la structure des débats (périodes législatives, sessions, sujets, discours, etc.), leurs métadonnées (mandats, titres, organes parlementaires, lieux, dates et heures), les intervenants (sexe, date de naissance, formation, appartenance à un parti, etc.), les partis politiques (nom(s), histoire, relations), les discours (orateur, texte, incidents, etc.), et l'annotation linguistique (normalisation, syntaxe, entités nommées, etc.) et multimédia (audio, vidéo, etc.). Pour répondre à ces besoins d'encodage, ils ont eu recours à plusieurs sous-groupes de la TEI, appelés modules :

— « TEI Transcription of speech » (spoken) pour les éléments du discours ;
— « TEI Common core » (core) et « TEI Metadata for Language Corpora » (corpus) pour la structure générale et les métadonnées ;
— « TEI Names, Dates, People and Places » (namesdates) pour les informations liées aux personnes, lieux et dates ;
— « TEI Linking, Segmentation and Alignment » (linking) pour élaborer des références ;
— « TEI Analysis and Interpretation » (analysis) et « TEI Feature structures » (iso-fs) pour l'annotation linguistique simple et complexe. [33]

L'intérêt de ces deux projets pour AGODA

Analyser ces deux projets d'encodage était une évidence pour notre projet AGODA. En effet, comme ParlaClarin et ParlaMint portaient sur une source similaire à la nôtre, et qu'ils proposaient une standardisation de l'encodage en XML-TEI pour cette dernière, il était essentiel pour nous de les prendre en compte et de s'en inspirer. Afin de nous inscrire dans cette lignée de standardisation, et en concertation avec l'équipe, j'ai donc analysé longuement les recommandations textuelles et les schémas de ces derniers et étudié de façon approfondie leurs corpus annotés, afin d'évaluer les choix effectués et de voir s'ils étaient applicables à notre corpus.

Les débats parlementaires étant propres à un pays et à une période donnés, nous avons cependant rencontré quelques différences entre les débats sur lesquels nous travaillions, et ceux traités par ces deux projets. Même si la retranscription des débats français actuels reste proche de celle produite sous la III^{e} République [34], certains éléments diffèrent.

32. Clarin (Éric), *ParlaMint*, GitHub, URL : `https://github.com/clarin-eric/ParlaMint` (visité le 14/08/2022).
33. Erjavec (Tomaž) et Pančur (Andrej), *Parla-CLARIN : TEI Guidelines for Corpora of Parliamentary Proceedings*, 19 sept. 2019, URL : `https://zenodo.org/record/3446164` (visité le 14/08/2022).
34. Puren, *et al.*, « Between History and Natural Language Processing », op. cit.

En effet, la publication des débats dans le *Journal officiel* de la III[e] République présente des caractéristiques éditoriales particulières, et des parties complémentaires telles que les annexes qui ne sont pas traitées par ParlaClarin et ParlaMint. Nous ne pouvions donc pas réutiliser en tout point leurs recommandations d'encodage. Nous ne souhaitions pas non plus réaliser une analyse linguistique à ce stade du projet. Nous avons donc retenu seulement une partie de leurs idées d'encodage répondant à nos objectifs, des choix concernant notamment la structure, les éléments du discours, les personnes et les lieux, et les métadonnées, tout en les adaptant à nos besoins spécifiques. Ces deux projets nous ont donc été très utiles pour orienter nos choix d'encodage.

4.2.3 Consulter les *guidelines* de la TEI

En outre, nous avons consulté les recommandations mises en place par le Consortium TEI, puisque ces dernières permettent en effet de « guider les pratiques d'encodage [35] ». La documentation, disponible en ligne [36] et en version imprimée, propose des « recommandations », et ne constitue pas un standard contraignant les pratiques. Ces recommandations apportent des conseils détaillés sur l'utilisation recommandée des balises TEI, et reposent sur l'expérience accumulée par la communauté TEI en matière de stratégie d'encodage. Cette documentation a, de plus, pour particularité de présenter un modèle abstrait, détaillant des concepts et leur relation entre eux [37].

Les recommandations reposent sur une documentation riche et structurée. Elles sont composées de vingt-trois chapitres, permettant de documenter les différents modules. Ces chapitres abordent d'abord des questions générales sur l'encodage de tout type de texte en TEI (chapitres 1 à 5), présentent ensuite les particularités d'encodage propre au type des textes (chapitres 6 à 12), et divers sujets concernant des applications spécialisées (chapitres 13 à 21), et mentionnent, enfin, des conseils pour la représentation du schéma TEI, et une définition de la notion de conformité (chapitres 22 et 23). Ces vingt-trois chapitres sont complétés par des propos liminaires portant sur des sujets généraux (versions, *Guidelines*, XML, etc.) et des annexes permettant notamment de lister les différents composants de la TEI (classes, éléments, attributs, macros, etc.).

35. Burnard, *Qu'est-ce que la Text Encoding Initiative ?*, op. cit.
36. *TEI : Text Encoding Initiative*, op. cit.
37. Albouy, « Les TEI Guidelines - Cours », op. cit.

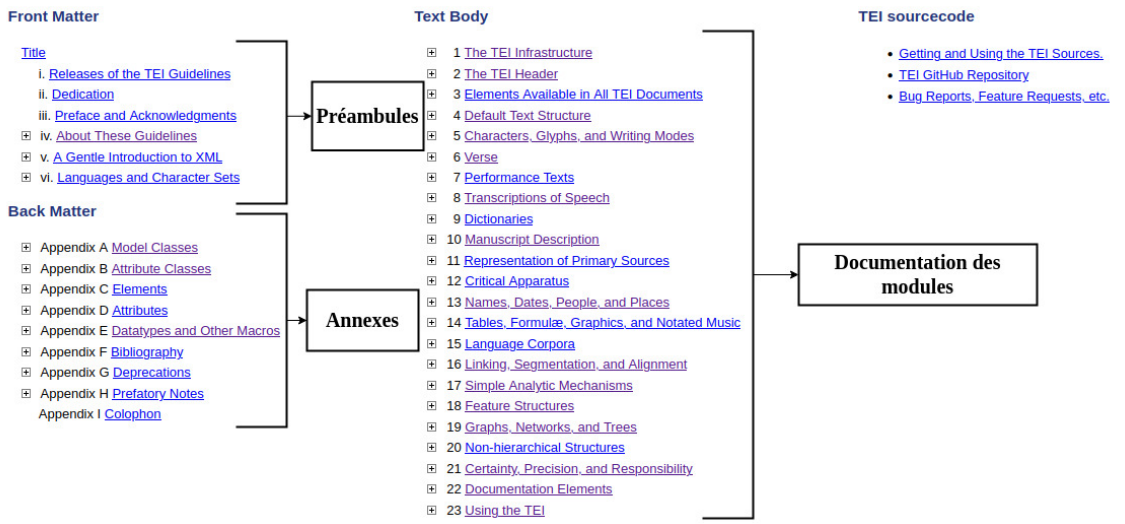
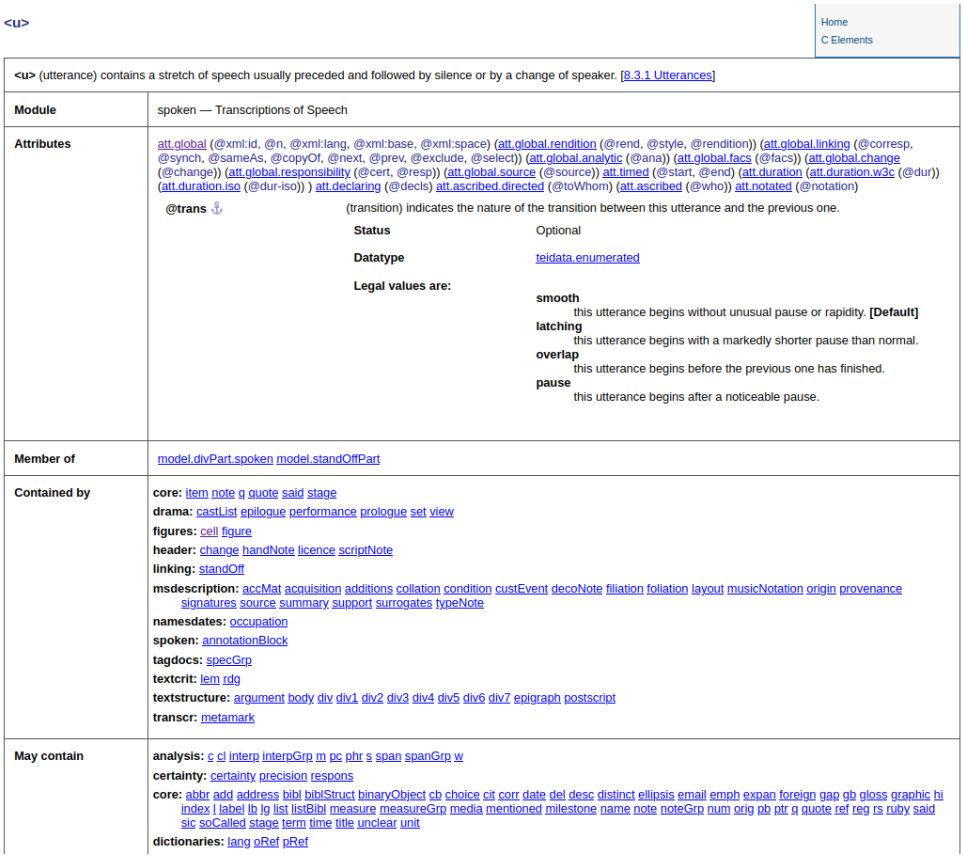


FIGURE 4.4 – Sommaire en ligne des *guidelines* de la TEI

Outre la documentation des modules, il est également possible d'accéder à des fiches descriptives des éléments. Ces dernières mettent à disposition une définition de l'élément, et précisent le module auquel il se rattache, les attributs qui peuvent lui être associés, et son contexte d'utilisation. Ces concepts sont illustrés aussi à l'aide de nombreux exemples d'utilisation de l'élément en question.



<u>

Home
C Elements

<u> (utterance) contains a stretch of speech usually preceded and followed by silence or by a change of speaker. [8.3.1 Utterances]

Module	spoken — Transcriptions of Speech
Attributes	att.global (@xml:id, @n, @xml:lang, @xml:base, @xml:space) (att.global.rendition (@rend, @style, @rendition)) (att.global.linking (@corresp, @synch, @sameAs, @copyOf, @next, @prev, @exclude, @select)) (att.global.analytic (@ana)) (att.global.facs (@facs)) (att.global.change (@change)) (att.global.responsibility (@cert, @resp)) (att.global.source (@source)) att.timed (@start, @end) (att.duration (att.duration.w3c (@dur)) (att.duration.iso (@dur-iso))) att.declaring (@decls) att.ascribed.directed (@toWhom) (att.ascribed (@who)) att.notated (@notation) **@trans** ⚓ (transition) indicates the nature of the transition between this utterance and the previous one. **Status** Optional **Datatype** teidata.enumerated **Legal values are:** **smooth** this utterance begins without unusual pause or rapidity. **[Default]** **latching** this utterance begins with a markedly shorter pause than normal. **overlap** this utterance begins before the previous one has finished. **pause** this utterance begins after a noticeable pause.
Member of	model.divPart.spoken model.standOffPart
Contained by	**core:** item note q quote said stage **drama:** castList epilogue performance prologue set view **figures:** cell figure **header:** change handNote licence scriptNote **linking:** standOff **msdescription:** accMat acquisition additions collation condition custEvent decoNote filiation foliation layout musicNotation origin provenance signatures source summary support surrogates typeNote **namesdates:** occupation **spoken:** annotationBlock **tagdocs:** specGrp **textcrit:** lem rdg **textstructure:** argument body div div1 div2 div3 div4 div5 div6 div7 epigraph postscript **transcr:** metamark
May contain	**analysis:** c cl interp interpGrp m pc phr s span spanGrp w **certainty:** certainty precision respons **core:** abbr add address bibl biblStruct binaryObject cb choice cit corr date del desc distinct ellipsis email emph expan foreign gap gb gloss graphic hi index l label lb lg list listBibl measure measureGrp media mentioned milestone name note noteGrp num orig pb ptr q quote ref reg rs ruby said sic soCalled stage term time title unclear unit **dictionaries:** lang oRef pRef

(a) Fiche descriptive de la balise <u>

Exemple

```
<u who="#fr_spkr1">si tu te déplaçais</u>
<u trans="latching" who="#fr_spkr2">Joe et moi l'aurions mis entre nous</u>
<list type="speakers">
 <item xml:id="fr_spkr1"/>
 <item xml:id="fr_spkr2"/>
</list>
```

Toute la liste ⚓

(b) Exemple d'application de la balise <u>

FIGURE 4.5 – Fiche descriptive et exemple d'utilisation de la balise <u>

Notre consultation des recommandations n'a pas consisté en une lecture complète, du début à la fin, de la documentation. En effet, comme nous venons de l'exposer, elles sont structurées de façon à ce que les utilisateurs puissent les parcourir facilement en fonction de leurs besoins généraux et spécifiques. Nous avons donc navigué selon nos besoins dans le sommaire de ces dernières. La richesse de cette documentation nous a aidés dans le choix et l'utilisation des balises. Nous avons pu analyser les possibilités d'encodage de notre corpus au regard des propositions de celle-ci. Enfin, cette démarche était fondamentale puisque nous voulions élaborer un encodage conforme à la TEI, en utilisant les balises de manière correcte du point de vue scientifique et technique.

4.3 Faire des choix

Au cours de la modélisation de l'encodage, j'ai réalisé, en concertation avec les chercheurs, un encodage test à la main [38], afin de tester nos choix, et de s'assurer de la conformité du modèle en construction. Pour cela, j'ai utilisé l'éditeur XML oXygen [39]. Et nous avons décidé d'appliquer l'encodage sur le compte rendu du débat de la séance du 26 novembre 1889, car il permettait de traiter un bon nombre des caractéristiques textuelles du corpus. Cet encodage nous a permis de nous rendre compte d'abord de certains dysfonctionnements, puis de concrétiser nos choix.

4.3.1 Les problématiques rencontrées lors de l'encodage test [40]

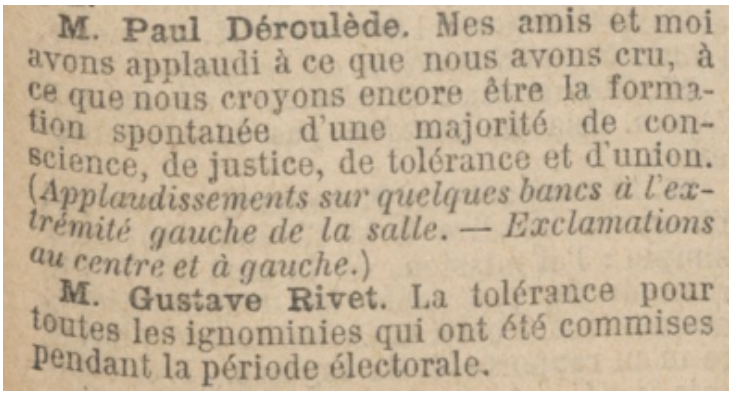
Lors de l'encodage test, nous avons été confrontés à plusieurs problématiques. L'encodage test est passé par plusieurs étapes, avant d'atteindre sa forme actuelle. Si sa conception a été aussi longue et laborieuse, c'est d'une part parce que les particularités de la source étaient complexes. Bien que les comptes rendus aient une structure homogène, leur contenu est hétéroclite, nécessitant de prendre en compte de nombreux éléments différents (organisation de la séance, discours rapportés, etc.). Leur contenu est aussi complexe par leur style (transformation du style oral vers le style écrit), complexifiant la réflexion sur les choix de balises. D'autre part, l'application des objectifs d'encodage, fixés en amont de la modélisation, représentait un défi, car il fallait trouver une solution pour les faire coexister au sein d'un même encodage. En outre, certaines problématiques ont été découvertes sur le tard, entraînant alors une remodélisation de l'encodage, ce qui a eu un impact sur la suite des missions à réaliser [41]. La suite de l'exposé relate les problématiques principales rencontrées.

La mise en page

La question de la conservation de la mise en page a été centrale tout au long de la modélisation. La publication des débats parlementaires ayant des caractéristiques éditoriales qui lui sont propres (texte divisé en colonnes, changements de pages, etc.), nous avons réfléchi initialement à l'utilité scientifique de les encoder. Nous avons décidé, suite à cela, de les prendre en compte, à l'exception des particularités typographiques (italique, gras, capitale). Ces aspects éditoriaux n'étant pas traités par les projets ParlaClarin et ParlaMint, nous avons dû sélectionner des balises de type formel et les agencer avec les autres balises de types structurel et sémantique. Cependant, cet agencement s'est avéré

38. Se référer à l'annexe B.2.
39. *Oxygen XML Editor*, URL : `https://www.oxygenxml.com/` (visité le 14/08/2022).
40. Pour illustrer les problématiques, se référer aux différentes versions de travail XML présentes en annexe B.1.
41. Se référer à la partie III pour le détail.

difficile puisque certaines balises propres à la mise en page ne pouvaient pas être alliées à certaines balises sémantiques, ce qui nous a amenés à nous requestionner sur la conservation de ces balises formelles. En effet, les comptes rendus des débats parlementaires relatent, en plus des discours, les événements perturbant les séances (applaudissements, interventions, etc.). Comme ParlaClarin et ParlaMint, nous avons voulu les encoder à l'aide de la balise TEI <incident>. Cette dernière, issue du module « spoken », permet justement d'encoder ce type d'élément. Mais l'intégration des changements de page (balise <pb>), des changements de colonnes (balise <cb>) et des sauts de ligne (balise <lb>) dans l'encodage a posé problème. En effet, l'incident pouvait être exposé sur plusieurs lignes, et chevaucher par la même deux colonnes ou deux pages. Pour ces cas-ci, il était impossible d'inclure les éléments <pb>, <cb> et <lb> au sein de l'élément <incident>, car la TEI ne le permettait pas.



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. (*Applaudissements sur quelques bancs à l'extrémité gauche de la salle. — Exclamations au centre et à gauche.*)

M. Gustave Rivet. La tolérance pour toutes les ignominies qui ont été commises pendant la période électorale.

(a) Source numérisée

```
<lb/><u who="#pers_ID" xml:id="CR_1889-11-26_u5" ana="#speaker">
   <seg xml:id="CR_1889-11-26_u5.1">
      <persName ref="#pers_ID">M. Paul Déroulède</persName>. Mes amis et moi
      <lb/>avons applaudi à ce que nous avons cru, à
      <lb/>ce que nous croyons encore être la forma-
      <lb/>tion spontanée d'une majorité de con-
      <lb/>science, de justice, de tolérance et d'union.
      <lb/><incident><desc>(Applaudissements sur quelques bancs à l'ex-
      <!-- Pas de lb possible dans incident --> trémité gauche de La salle. – Exclamations
      <!-- Pas de lb possible dans incident --> au centre et à gauche.)</desc></incident>
   </seg>
</u>

<lb/><u who="pers_ID" xml:id="CR_1889-11-26_u6" ana="#speaker">
   <seg xml:id="CR_1889-11-26_u6.1">
      <persName ref="#pers_ID">Gustave Rivet</persName>. La tolérance pour
      <lb/>toutes Les ignominies qui ont été commises
      <lb/>Pendant la période électorale.
   </seg>
</u>
```

(b) Modèle d'encodage prenant en compte la mise en page

FIGURE 4.6 – Extrait d'une prise de parole lors de la séance parlementaire du 26 novembre 1889

Outre le cas des événements perturbants, l'intégration de ces balises formelles dans l'encodage des annexes était aussi complexe.

Face à ces difficultés liées aux contraintes de la TEI, nous avons requestionné nos objectifs et avons abandonné l'idée d'encoder les informations de mise en page du journal. Mais nous avons très vite laissé de côté cette solution, car les changements de page représentaient une information essentielle pour la question du sourçage.

Nous avons donc opté pour une solution médiane incluant les éléments de mise en page que nous jugions essentiels. Nous avons donc mis de côté les sauts de lignes et les changements de colonnes, car ces éléments n'apportaient pas d'intérêt majeur pour nos objectifs d'encodage, et nous avons conservé le balisage des changements de page. Nous avons donc dû trouver une solution pour intégrer cet élément au sein de l'encodage. La première idée adoptée n'était pas la plus optimale. Nous voulions utiliser l'élément <floatingText>, car il pouvait être inclus dans les balises <incident>. Mais il devait contenir un certain nombre d'éléments obligatoires avant de pouvoir préciser le changement de page lui-même.

```
<floatingText>
   <body>
      <div>
         <pb n="177"/>
      </div>
   </body>
</floatingText>
```

FIGURE 4.7 – Gestion des changements de page à l'aide de l'élément <floatingText>

Après de nombreux essais, et grâce au recours de la liste TEI[42], nous avons fait le choix de simplifier l'encodage en employant l'élément <pb> pour tous les cas où le changement de page ne s'effectue pas au sein d'un incident, et l'élément <ref> pour le cas contraire.

```
<pb n="#187"/>

<!-- ... -->

<incident>
   <desc>(Bruit <ref target="#188"/> à gauche)</desc>
</incident>

<!-- ... -->
```

FIGURE 4.8 – Gestion des changements de page à l'aide des éléments <pb> et <ref>

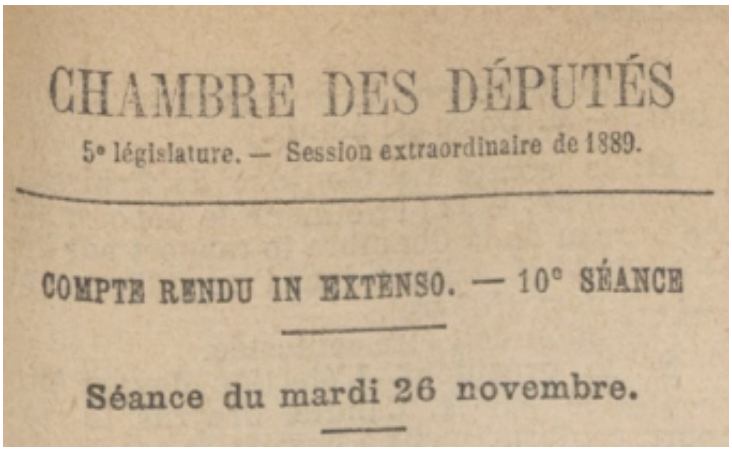
Cette solution est pour l'instant celle qui a été adoptée. Mais une autre solution pourrait être appliquée dans le futur, consistant à séparer les encodages formel et sémantique. L'encodage formel pourrait inclure l'information des changements non pas par la balise <pb> comme évoqué, mais par un système permettant de relier directement dans

42. La liste TEI est un lieu d'échange, à l'échelle de la communauté, permettant d'aider à définir les « meilleures pratiques » en fonction des besoins des utilisateurs.

l'encodage le lien de la page numérisée avec le texte encodé. Cela pourrait se faire à l'aide de l'élément <facsimilé> et <surface> pour indiquer les coordonnées spatiales de l'image et l'attribut @facs pour associer cette zone au texte. Le numéro de page étant indiqué sur la numérisation, il serait alors d'emblée joint au texte. Mais cette idée, apparue que tardivement, a été mise de côté, car elle nécessitait de remodéliser l'encodage, une tâche que nous ne pouvions pas nous permettre au vu du processus d'automatisation de l'encodage déjà initié.

La manchette

La manchette du journal correspond, comme nous l'avons vu précédemment, au titre de la publication, celui-ci contenant des informations importantes telles que le nom de l'Assemblée, le numéro de législature, le type de session et sa date, le numéro et la date de la séance. Nous nous sommes questionnés sur l'encodage de celle-ci. En effet, ces dernières avaient un double enjeu : elles correspondaient à une partie du texte à encoder, mais elles constituaient aussi des informations fondamentales pour les métadonnées. Nous avons souhaité, en premier lieu, encoder une partie de ces informations au sein de l'élément <front>, celui-ci permettant de contenir tout ce qui est au début d'un document, avant le corps même du texte. Cet élément était situé juste au-dessus de l'élément <body> contenant le texte.



(a) Source numérisée

```
<text>

   <front>
      <div type="preface">
         <head>COMPTE RENDU IN EXTENSO. – <num>10</num>e SÉANCE</head>
         <head>Séance du <date>mardi 26 novembre</date>.</head>
      </div>
   </front>

   <body>
      <!-- ... -->
```

(b) Modèle d'encodage construit à l'aide de l'élément <front>

FIGURE 4.9 – Manchette du compte rendu de la séance parlementaire du 26 novembre 1889

Mais nous avons finalement abandonné cette idée, car l'élément <front> en lui-même n'apportait rien à l'encodage, si ce n'est un détail formel du journal. De plus, comme cet en-tête correspondait à des informations constitutives des métadonnées, le garder au sein du texte avec la balise <front> entraînait une duplication des informations, ce qui n'était pas utile. L'en-tête présent dans la publication a donc été supprimé du texte à proprement parlé, et intégrée au sein de l'élément <teiHeader>.

Les parties complémentaires

Les parties complémentaires présentes à la fin des comptes rendus (annexes, erratum, rectifications, etc.) ont également constitué une difficulté importante. Nous nous sommes demandés s'il fallait les intégrer au sein du texte, contenu dans l'élément <body>, ou au contraire, les distinguer du corps du texte en les comprenant dans l'élément <back>, prévu pour l'encodage des suppléments placés après la partie principale d'un texte. Cette incertitude était due au manque d'informations quant à la composition de ces parties complémentaires. En effet, nous n'arrivions pas à déterminer si elles représentaient des

propos tenus lors des séances, et donc si elles faisaient partie intégrante du corps du texte, ou si elles étaient incluses par la suite, en complément du compte rendu. Si le cas des annexes était clair, celui des autres parties rencontrées ne l'était pas. Nous avions donc pensé initialement encoder seulement les annexes au sein de l'élément <back>. Mais après avoir consulté plusieurs références sur la conception des comptes rendus, et suite à une analyse plus fine de la source, nous avons modifié notre choix initial, et avons inclus chacune des parties complémentaires dans l'élément <back>, au même titre que les annexes. En effet, notre analyse nous a confirmé que ces parties étaient introduites en complément du compte rendu lui-même.

En plus de ce questionnement, l'encodage spécifique des annexes a constitué une difficulté majeure, due à des problèmes d'agencement de balises au même titre que pour la mise en page. En effet, comme les annexes contiennent différents éléments tels que le détail des scrutins, la liste des votants classés en fonction de leur vote, et des rectifications de scrutins de la séance précédente, nous devions créer un encodage permettant de tous les prendre en compte. ParlaClarin et ParlaMint n'ayant pas d'annexes similaires aux nôtres, nous avons dû faire divers essais, avant d'obtenir un résultat valide.

La gestion du corpus

La gestion du corpus a été un enjeu important. Bien que l'encodage test n'ait été réalisé que pour un seul débat, nous avions conscience qu'il fallait pouvoir gérer l'ensemble des débats du corpus, en les encodant de la même manière et en les regroupant afin de préserver la cohérence du corpus. Nous nous sommes demandés alors comment les rassembler techniquement : fallait-il les regrouper au sein d'un même fichier XML ou sous la forme de plusieurs fichiers XML ? Nous nous sommes également questionnés sur quels débats réunir : fallait-il garder le corpus initial, regroupant l'ensemble des débats pris en compte par le projet AGODA, ou fallait-il créer des sous-corpus tenant compte par exemple de l'année ou du type de session ? La difficulté de ces questionnements reposait aussi sur la liaison des débats entre eux. En effet, certaines informations pouvaient porter sur un débat antérieur (rectifications de vote, etc.) ou postérieur (l'ordre du jour, etc.). Il fallait donc pouvoir le mentionner au sein de l'encodage, et créer ainsi une liaison entre deux débats distincts. Nous avons donc mis en place le système suivant : nous avons décidé de créer un fichier central XML, permettant de regrouper un ensemble de fichiers XML composants. Ce fichier central est construit à l'aide de l'élément racine <teiCorpus>, il contient des métadonnées propres aux spécificités du corpus et stocke l'ensemble des fichiers qui le compose à l'aide du mécanisme d'inclusion XInclude. Le composant, quant à lui, contient ses propres métadonnées, et le texte des débats. Il regroupe un seul numéro du journal à la fois. Ce système nous permettait d'organiser au mieux l'ensemble des fichiers, mais aussi de rendre possible la liaison entre les débats, lorsque cela était nécessaire. Par exemple, nous avons pu relier le résultat du vote annoncé dans le compte rendu de la

séance du 25 avec les rectifications apportées à ces votes lors de la séance du 26. Nous avons relié ces deux informations à l'aide d'un identifiant unique inclus par l'attribut @corresp. La reconnaissance de l'identifiant, répété entre les deux fichiers, était rendue possible, car les deux fichiers XML étaient reliés à l'aide du mécanisme d'inclusion.

Par ailleurs, nous avons fait le choix qu'un corpus correspondrait à un ensemble de débats propres à une législature. Ce regroupement a été choisi puisque le projet AGODA souhaitait travailler par législature, et que l'objet de mon stage portait également sur une seule législature. La justification de ce choix reposait aussi sur les caractéristiques textuelles des débats, les publications pouvant différer légèrement d'une législature à l'autre. Ces modifications auraient pu alors avoir un impact sur le modèle d'encodage et auraient nécessité une mise à jour de ce dernier. Il était alors important pour nous d'être précautionneux à ce sujet.

4.3.2 Résultat de la modélisation [43]

Le résultat obtenu de l'encodage test est le fruit de tous les analyses et essais évoqués précédemment. Les choix sont fixes, mais pourront être amenés à évoluer dans la suite du projet. Nous allons exposer les principales caractéristiques du modèle d'encodage propres au fichier composant, puis propres au fichier corpus, tout en évoquant enfin les améliorations possibles de la modélisation.

L'encodage prend en compte un balisage physique et sémantique de la source. Le balisage physique correspond à l'encodage de la structure du compte rendu, comprenant deux catégories structurelles différentes : d'une part, les éléments structurels formels (mise en page), d'autre part, les éléments structurels logiques (organisation du contenu textuel). Le balisage sémantique, lui, correspond à un premier niveau d'encodage analytique, tenant compte des éléments constitutifs du discours et des entités nommées.

Structure générale du composant

Chaque composant est un document XML. Il correspond à l'encodage d'un seul numéro du *Journal officiel*, celui-ci pouvant contenir un ou des compte(s) rendu(s) de séance(s). Ce document a pour racine l'élément <TEI>. Il contient ensuite :

— l'élément enfant <teiHeader>, permettant d'indiquer les métadonnées spécifiques du composant en question ;

— l'élément enfant <text>, contenant le(s) compte(s) rendu(s) dans l'élément <body>, ainsi que les parties complémentaires incluses dans l'élément <back>.

Le corps du texte <body> peut comporter un ou plusieurs compte(s) rendu(s), c'est pourquoi nous l'avons divisé en plusieurs divisions, selon le nombre de comptes rendus

43. L'intérêt de cette partie n'est pas de dupliquer l'intégralité de la documentation textuelle de l'encodage, elle présente juste les principales caractéristiques du modèle défini. Pour plus d'informations, se référer directement à : Lebreton, Puren et Vernus, *AGODA*, op. cit.

présents. Pour le premier compte rendu, l'élément est complété par l'attribut @type ayant pour valeur « sitting », tandis que chacun des autres comptes rendus prennent pour valeur « other-sitting ».

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="FR_3R_5L_1889-11-26" xml:lang="fr">

   <!-- Métadonnées du composant -->
   <teiHeader>
      <fileDesc>
         <titleStmt>
            <title></title>
         </titleStmt>
         <publicationStmt>
            <p></p>
         </publicationStmt>
         <sourceDesc>
            <p></p>
         </sourceDesc>
      </fileDesc>
   </teiHeader>

   <!-- Compte(s) rendu(s) -->
   <text ana="#FR_3R_5L_1889-11-26">

      <!-- Parties principales -->
      <body>
         <!-- Premier compte rendu -->
         <div type="sitting"><!--...--></div>

         <!-- Deuxième compte rendu -->
         <div type="other-sitting"><!--...--></div>
      </body>

      <!-- Parties complémentaires -->
      <back><!--...--></back>

   </text>
</TEI>
```

FIGURE 4.10 – Structure générale du fichier composant

Les métadonnées du composant

Les métadonnées sont contenues dans l'élément <teiHeader>, prévu à cet effet. Il est ensuite divisé selon la fonction de ces dernières :

— l'élément <fileDesc> correspond à la description bibliographique du fichier électronique. Il contient :

- les métadonnées bibliographiques du fichier électronique divisé en sous-éléments (mentions de titre, taille du corpus, mentions de publication) ;
- les informations décrivant la source dont le fichier XML-TEI est dérivé, ces dernières étant inclus dans l'élément <sourceDesc>, lui-même contenant l'élément <biblFull> divisé en sous-éléments (mentions de titre, mentions de publication, mentions de collection).

```
<fileDesc>
   <titleStmt>
      <title type="main" xml:lang="fr">Journal officiel de la République française. Débats
         parlementaires</title>
      <title type="main" xml:lang="en">Official Journal of the French Republic. Parliamentary
         debates</title>
      <title type="sub" xml:lang="fr">Chambre des députés : compte rendu in-extenso</title>
      <title type="sub" xml:lang="en">Chamber of Deputies: verbatim report</title>
      <meeting n="E1" ana="#parla.lower #parla.session">Session extraordinaire de
         1889</meeting>
      <meeting n="5L" ana="#parla.lower #parla.legislature">5e législature</meeting>
      <meeting n="10" ana="#parla.lower #parla.sitting">10e séance</meeting>
      <respStmt>
         <persName>
            <forename>Fanny</forename>
            <surname>Lebreton</surname>
            <ptr type="id-hal" target="fanny-lebreton"/>
         </persName>
         <resp xml:lang="fr">Transformation du JSON en XML-TEI et ajout automatique des
            balises TEI par des scripts Python</resp>
         <resp xml:lang="en">Transformation from JSON to XML-TEI and automatic addition of TEI
            tags by Python scripts</resp>
      </respStmt>
      <respStmt>
         <persName>
            <forename>Marie</forename>
            <surname>Puren</surname>
            <ptr type="id-hal" target="marie-puren"/>
            <ptr type="orcid" target="0000-0001-5452-3913"/>
         </persName>
         <resp xml:lang="fr">TEI Header</resp>
         <resp xml:lang="en">TEI Header</resp>
      </respStmt>
      <funder>
         <orgName xml:lang="fr">Bibliothèque nationale de France</orgName>
         <orgName xml:lang="en">National Library of France</orgName>
      </funder>
   </titleStmt>
```

FIGURE 4.11 – Extrait de l'élément <fileDesc> du fichier composant

— l'élément <encodingDesc> permet de documenter les choix d'encodage. Il contient :
- le contexte de production des fichiers XML-TEI (élément <projectDesc>) ;
- les informations sur la fréquence d'apparition des balises de l'encodage (élément <tagsDecl>).

```
<encodingDesc>
   <projectDesc>
      <p xml:lang="fr"><ref target="https://www.bnf.fr/fr/les-projets-de-recherche#bnf-agoda"
         >AGODA</ref> est un projet qui a pour objectif rendre disponible au format XML-TEI
      les textes de débats parlementaires à la Chambre des députés au cours de la Troisième
      République, suivant l'<ref
         target="https://github.com/mpuren/agoda/blob/ODD/documentation/agoda_odd.xml"
         >ODD</ref> défini pour le projet à partir des <ref
         target="https://github.com/clarin-eric/parla-clarin">recommandations produites par
         Parla-CLARIN</ref>. Dans une optique de preuve de concept, la phase 1 du projet
      AGODA se concentre plus particulièrement sur la 5ème législature (1889-1893). Les
      textes encodés sont d'abord extraits des documents numérisés disponibles sur <ref
         target="https://gallica.bnf.fr/ark:/12148/cb328020951/date.item">Gallica</ref>, la
      bibliothèque numérique de la Bibliothèque nationale de France, puis ils sont convertis
      en XML-TEI au moyen de scripts Python.</p>
      <p xml:lang="en">is a project that aims to make available in XML-TEI format the texts of
      parliamentary debates in the Chamber of Deputies during the Third Republic, following
      the <ref
         target="https://github.com/mpuren/agoda/blob/ODD/documentation/agoda_odd.xml"
         >ODD</ref> defined for the project from the <ref
         target="https://github.com/clarin-eric/parla-clarin">Parla-CLARIN
         recommendations</ref>. From a proof-of-concept perspective, phase 1 of the AGODA
      project focuses more specifically on the 5th legislature (1889-1893). The encoded
      texts are first extracted from the digitised documents available on <ref
         target="https://gallica.bnf.fr/ark:/12148/cb328020951/date.item">Gallica</ref>,
      the digital library of the Bibliothèque nationale de France, then they are converted
      into XML-TEI using Python scripts.</p>
   </projectDesc>
```

FIGURE 4.12 – Extrait de l'élément <encodingDesc> du fichier composant

— l'élément <profileDesc> permet de détailler les informations non bibliographiques de la source. Il contient :
- la langue du texte (élément <langUsage>) ;
- le contexte dans lequel ont lieu les débats (élément <settingDesc>).

```
<profileDesc>
   <langUsage>
      <language ident="fr">Français</language>
   </langUsage>
   <settingDesc>
      <setting>
         <name type="place">Palais Bourbon</name>
         <name type="city">Paris</name>
         <name type="country" key="FR">France</name>
         <date when="1889-11-26">mardi 26 novembre</date>
      </setting>
   </settingDesc>
</profileDesc>
```

FIGURE 4.13 – Élément <profileDesc> du fichier composant

Le balisage formel

Le balisage formel ne prend pas en compte les alinéas, les sauts de ligne, les changements de colonne, l'emplacement du texte, et les éléments typographiques. Le balisage formel se situe alors sur les éléments suivants :

— les changements de page ;

```
<!-- Changement de page -->
<pb n="175"/>

<!-- Changement de page au sein d'un incident -->
<incident>
   <desc> (Bruit <ref target="#188"/> à gauche) </desc>
</incident>
```

FIGURE 4.14 – Modèle d'encodage des changements de page

— les paragraphes (numérotés) ;

```
<!-- Paragraphe -->
<seg xml:id="CR_1889-11-26_u1.1">
   <persName ref="#pers_ID">M. Paul
      Déroulède</persName>. Je demande la parole.
</seg>
```

FIGURE 4.15 – Modèle d'encodage des paragraphes

— la signature du chef du service sténographique ;

```
<!-- Signature -->
<signed>
   <seg>Le <roleName ref="#pers_ID">Chef du service
      sténographique</roleName> de la <orgName ref="#org_ID">Chambre des députés</orgName>, <persName ref="#pers_ID">EMILE GROSSELIN</persName>.</seg>
</signed>
```

FIGURE 4.16 – Modèle d'encodage de la signature

Le balisage logique

Le balisage logique comprend l'encodage du sommaire, du corps du texte, et des parties complémentaires. Ces derniers ont chacun des particularités qui leur sont propres :

— le sommaire correspond à une division étant caractérisé par l'attribut @type ayant pour valeur « contents ». Il contient :

- un en-tête (élément <head>) ;
- une liste (élément <list>) contenant des items (élément <item>) correspondant aux différentes parties de la séance.

```
<div type="contents">
   <head>SOMMAIRE</head>
   <list>
      <item xml:id="pv">Procès verbal : MM. <persName ref="#PD_2409">Paul Déroulède</persName>, <persName ref="#pers_ID">Georges Laguerre</persName>,
         <persName ref="#pers_ID">Briens</persName>,<persName ref="#pers_ID">Bizouard-Bert</persName>,<persName ref="#pers_ID">Vernière</persName>.</item>
      <item><!-- ... --></item>
      <!-- ... -->
   </list>
</div>
```

FIGURE 4.17 – Modèle d'encodage du sommaire

— le corps du texte est composé de plusieurs parties incluses dans des sous-divisions caractérisées par l'attribut @type ayant pour valeur « part ». Les parties non mentionnées dans le sommaire sont caractérisées avec une valeur de l'attribut @type plus précise. Ces parties peuvent contenir :

- un titre (élément <head>) ;
- des notes structurelles (élément <note>) ;
- des tableaux (élément <table>).

```
<div type="part" corresp="#àdéfinir">
   <head>COMMUNICATION DU GOUVERNEMENT.</head>
   <!-- [...] -->
</div>
```

FIGURE 4.18 – Modèle d'encodage des parties du corps du texte

— les parties complémentaires sont incluses dans des sous-divisions caractérisées à l'aide de l'attribut @type ayant pour valeur la fonction de la partie. Ces dernières se divisent de la façon suivante :

- toutes les parties complémentaires or annexes gardent la même structure que le corps du texte ;
- les annexes, quant à elle, sont structurées de façon particulière. Chaque scrutin est inclus dans une division et contient un titre développé (élément <head> contenant les éléments <label> et <note>), le détail des scrutins (élément <desc> pour le détail des votes, et <note> pour le résultat), la liste des votants classés selon leur vote (élément <note> caractérisé par l'attribut @type de valeur « voterslist »), et enfin de possibles corrections des votes de la séance même, et des rectifications des votes de la séance précédente.

```
<head>Annexes au procès-verbal de la séance du <date when="1889-11-26">mardi 26 novembre
   1889</date>.</head>

<div xml:id="CR_1889-11-26_vot">

   <!-- VOTE 1 -->
   <div xml:id="CR_1889-11-26_vot1" type="voting" corresp="#discussion7ebureau">

      <!-- Titre -->
      <head>
         <label>SCRUTIN</label>
         <note xml:id="CR_1889-11-26_n5"><seg xml:id="CR_1889-11-26_n5.1">Sur les
            conclusions du <num>7e</num> bureau tendant à l'annulation des opérations
            électorales de la <placeName ref="#lieu_ID"><num>1re</num> circonscription
               de l'arrondissement de Lorient (Morbihan)</placeName>.</seg></note>
      </head>

      <!-- Détail des scrutins -->
      <desc>
         <measure type="nbvoters" quantity="506">Nombre des votants
            <num>506</num></measure>
         <measure type="majority" quantity="254">Majorité absolue <num>254</num></measure>
         <measure type="ayes" quantity="330">Pour l'adoption <num>330</num></measure>
         <measure type="noes" quantity="176">Contre <num>176</num></measure>
      </desc>

      <!-- Résultat des scrutins -->
      <note type="result" xml:id="CR_1889-11-26_n6"><seg xml:id="CR_1889-11-26_n6.1">La
         <orgName ref="#org_ID">Chambre des députés</orgName> a adopté.</seg></note>

      <!-- Liste des votants -->
      <note type="voterslist" xml:id="CR_1889-11-26_n7">
         <desc>Ont voté pour :</desc>
         <seg xml:id="CR_1889-11-26_n7.1"> MM.<persName ref="#pers_ID">Abeille</persName>.
            <persName ref="#pers_ID">Arène (Emmanuel)</persName>. <persName
               ref="#pers_ID">Armez</persName>. <persName ref="#pers_ID"
                  >Arribat</persName>. <persName ref="#pers_ID">Audifred</persName>. <persName
                     ref="#pers_ID">Aynard (Edouard)</persName>. </seg>
         <!-- [...] -->
         <desc>Ont voté contre :</desc>
         <seg xml:id="CR_1889-11-26_n7.3"> MM. <persName ref="#pers_ID">Abrial
            (Léon)</persName>. <persName ref="#pers_ID">Adam (Achille)</persName>.
            <persName ref="#pers_ID">Aigle (comte de l')</persName>. <persName
               ref="#pers_ID">Ailliéres (d')</persName>. <persName ref="#pers_ID">Aimel
                  (Henri)</persName>. <persName ref="#pers_ID">Amagat</persName>. <persName
                     ref="#pers_ID">Arenberg (prince d')</persName>. <persName ref="#pers_ID"
                        >Argeliès</persName>. <persName ref="#pers_ID">Arnault</persName>. <persName
                           ref="#pers_ID">Arnous</persName>. </seg>
         <!-- [...] -->
      </note>
```

FIGURE 4.19 – Extrait du modèle d'encodage des annexes

Le balisage sémantique

Le balisage sémantique repose d'abord sur les éléments du discours. Ces derniers sont constitués des informations suivantes :

— les énoncés correspondent à la prise de parole d'une personne, chacun d'eux est inclus dans l'élément <u>, élément TEI permettant d'encoder une partie de discours, et est complété par l'attribut @who, permettant d'associer le nom d'une personne à la prise de parole à l'aide d'un identifiant unique, et de pointer vers les métadonnées de cette dernière, et l'attribut @ana permettant de pointer vers la typologie des types d'orateurs (président, secrétaire, députés, etc.) afin de distinguer les différentes prises de parole ;

```
<u who="#pers_ID" xml:id="CR_1889-11-26_ul"
   ana="#speaker"> ... </u>
```

FIGURE 4.20 – Modèle d'encodage d'une prise de parole

— les commentaires des sténographes, ces derniers indiquant l'heure du début et l'heure de la fin de la séance, les phénomènes ou événements interrompant le discours, les actions effectuées en séance (vote, dépouillement, etc.). Deux types de balisage ont été mis en place :

- le premier balisage repose sur les commentaires portant sur l'organisation de la séance et les actions effectuées, ils sont encodés à l'aide de l'élément <note>, caractérisés avec l'attribut @type prenant pour valeur la nature du commentaire ;

```
<note type="opening" xml:id="CR_1889-11-26_n1"><seg xml:id="CR_1889-11-26_n1.1">La
      séance est ouverte à <time when="02:00:00">deux heures</time>.</seg></note>
```

FIGURE 4.21 – Modèle d'encodage d'un commentaire avec l'élément <note>

- le deuxième balisage repose sur les commentaires portant sur l'atmosphère de la séance, ils sont encodés à l'aide de l'élément <incident> prévu à cet effet, et du sous-élément <desc>.

```
<incident>
   <desc>(Exclamations ironiques au centre et à gauche.)</desc>
</incident>
```

FIGURE 4.22 – Modèle d'encodage d'un commentaire avec l'élément <incident>

— les citations présentes dans les discours, chacune d'elle étant incluse dans l'élément <quote> prévu pour contenir une expression ou un passage dont l'origine est extérieure au texte.

```
<quote>« Oui, nous les validerons tous ! »</quote>
```

FIGURE 4.23 – Modèle d'encodage d'une citation

Le balisage sémantique repose ensuite sur la mise en évidence des entités nommées. Ces dernières sont constituées :

— des personnes, contenues dans l'élément <persName>, avec pour attribut @ref, celui-ci permettant de renvoyer vers la description de la personne présente dans l'index des noms de personnes citées. ;
— des lieux, contenus dans l'élément <placeName>, avec pour attribut @ref, celui-ci permettant de renvoyer vers l'index des lieux cités ;
— des organisations, contenues dans l'élément <orgName>, avec pour attribut @ref, celui-ci permettant de renvoyer vers l'index des organisations citées ;
— des éléments temporels (élément <date> pour les dates, élément <time> pour les heures) et éléments quantifiables (élément <num> pour les nombres).

```
<!-- Personne -->
<persName ref="#pers_ID">M. Leygues</persName>

<!-- Lieu -->
<placeName ref="#lieu_ID">Département de la Corrèze, arrondissement de Tulle, <num>1re</num> circonscription</placeName>

<!-- Organisation -->
<orgName ref="#org_ID">Chambre des députés</orgName>

<!-- Date, temps, nombre -->
<date when="1880-11-11">11 novembre 1880</date>
<time when="02:00:00">deux heures</time>
<num>12,189</num>
```

FIGURE 4.24 – Modèle d'encodage des entités nommées

Structure générale du corpus

Un fichier XML corpus permet de regrouper les comptes rendus des débats parlementaires propres à une législature. Il s'agit d'un document XML, ayant comme racine l'élément <teiCorpus>. Il contient ensuite :

— l'élément enfant <teiHeader>, permettant d'indiquer les métadonnées du corpus dans son ensemble ;

— l'élément enfant <standOff>, contenant les données liées et les informations contextuelles ;

— une série d'éléments <xi :include>, permettant de stocker l'ensemble des fichiers XML comportant les comptes rendus d'une même législature au sein de ce fichier XML central.

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0" xml:id="FR_3R_5L" xml:lang="fr">

   <!-- Métadonnées du corpus -->
   <teiHeader>...</teiHeader>
   <standOff>...</standOff>

   <!-- Stockage du composant correspondant à la séance du 26 novembre 1889 -->
   <xi:include xmlns:xi="http://www.w3.org/2001/XInclude" href="FR_3R_5L_1889-11-26.xml"/>


   <!-- Stockage des autres composants du corpus de façon identique -->
   ...
</teiCorpus>
```

FIGURE 4.25 – Structure générale du fichier corpus

Les métadonnées du corpus

Les métadonnées du fichier XML corpus concernent l'ensemble des documents du corpus. Elles sont construites de façon similaire à celles des fichiers composants, mais peuvent différer à certains endroits. Les principales différences reposent sur :

— les composants de l'élément <encodingDesc> :

- intégration de l'élément <editorialDecl>, indiquant les pratiques éditoriales ;
- intégration de l'élément <appInfo>, permettant de donner des informations sur les logiciels et les langages utilisés pour traiter les documents numérisés ;

- intégration de l'élément <classDecl>, détaillant les taxonomies [44] utilisées au sein des fichiers composants.

— les composants de l'élément <profileDesc> :
- intégration de l'élément <textClass>, précisant selon un système de classification standardisé la nature des textes du corpus ;
- intégration de l'élément <particDesc> contenant la liste descriptive des orateurs.

Le mécanisme d'inclusion

Les inclusions de fichiers sont réalisées à l'aide des éléments <xi :include>. Ces derniers sont construits à l'aide :

— de l'attribut @xmlns :xi prenant pour valeur l'espace de nom propre au mécanisme XInclude ;
— de l'attribut @href, indiquant le chemin menant vers le fichier XML à inclure.

4.3.3 Un modèle concluant ?

Améliorations du modèle

Le modèle d'encodage élaboré peut être approfondi sur plusieurs points. Tout d'abord, bien qu'il ait été pensé en fonction des nombreux objectifs d'encodage initiaux, il ne les prend pas tous en compte. Il s'agira d'intégrer, dans le futur, l'annotation linguistique, et d'associer au texte les sujets extraits grâce au *topic modeling* :

— Concernant l'annotation linguistique, il sera possible de s'inspirer des modèles d'encodage de ParlaClarin et ParlaMint, tous deux traitant cet aspect ;
— Concernant les sujets, ils pourront être inclus au sein de cette annotation linguistique, au niveau du mot (élément <w>). Chaque mot pourra être relié à son sujet grâce à l'identifiant unique du sujet en question. L'ensemble des *topics* pourront être rassemblés dans l'élément <standOff> du fichier corpus.

Ces deux enjeux pourront être inclus au sein même de l'encodage existant, ou faire l'objet d'un encodage à part. Cela reste à déterminer.

Par ailleurs, l'encodage test est lacunaire et nécessite d'être complété sur plusieurs points. En effet, comme nous avons été confrontés à un certain nombre de problématiques et que certaines ont été prises en compte tardivement, nous n'avons pas eu le temps de finaliser l'ensemble de l'encodage test. Il s'agira alors dans un premier temps de réaliser l'encodage complet du compte rendu, puisque seules les parties principales l'ont été. Dans un deuxième temps, il faudra compléter ce qui a été réalisé notamment au niveau des

44. Une taxonomie est une méthode de classification des informations. Dans notre cas, nous avons créé par exemple une taxonomie portant sur les types d'orateurs (orateur, invité, membre du gouvernement, inconnu).

valeurs d'attributs. Par exemple, nous n'avons pas eu le temps d'associer un identifiant unique à l'encodage des orateurs, provenant par exemple de data.bnf[45] , afin de s'inscrire dans le contexte du Web sémantique et des données liées[46] . Il pourra être intégré au sein de l'élément <person> (attribut @xml :id), présent au sein de l'élément <particDesc> (fichier corpus) contenant la liste des orateurs. Il pourra être ensuite associé aux orateurs dans les fichiers composants en y faisant référence grâce aux attributs @who (contenu dans l'élément <u>) et @ref (inclus dans l'élément <persName>). Cette démarche pourra être appliquée de la même façon aussi sur les autres entités nommées (lieux et organisations). L'encodage test du corpus pourra être complété, aussi, au niveau du <standOff>, celui-ci ayant été mis de côté, et sur le contenu des différentes taxonomies. Enfin, il serait utile de compléter le modèle d'encodage en réalisant un autre encodage test à la main sur un compte rendu présentant d'autres particularités. Bien que la séance du 26 novembre 1889 était relativement riche, elle ne comportait pas certaines particularités telles que le cas des comptes rendus relatant plusieurs séances, ou encore les parties complémentaires autres que celles présentent dans la séance traitée. Il serait intéressant alors d'encoder concrètement ces cas afin de pouvoir illustrer l'encodage idéal de ces derniers.

Même si ces diverses améliorations prendront du temps, elles permettront d'obtenir un modèle de balisage final et représenteront une aide précieuse pour les différentes étapes de l'automatisation de l'encodage.

45. data.bnf est une base de données donnant accès à des fiches de référence sur les auteurs, les œuvres et les thèmes. Cette dernière regroupe et lie des données issues de bases de données distinctes, produites dans des formats différents. Elle s'inscrit dans la perspective du web sémantique en adoptant des standards de données promus par le Consortium W3C et en les publiant selon ces standards. Site web disponible à l'adresse suivante : `https://data.bnf.fr/`.

46. Puren (Marie), Vernus (Pierre), Pellet (Aurélien), Bourgeois (Nicolas) et Lebreton (Fanny), « Le Projet AGODA. Annoter et Publier Les Débats Parlementaires Français de La Fin Du XIXe Siècle : Défis et Solutions », dans *Colloque Humanistica 2022*, Montréal, Canada, 2022, URL : `https://hal.archives-ouvertes.fr/hal-03674919` (visité le 09/08/2022).

Chapitre 5

Documenter et formaliser l'encodage

Une fois les grandes lignes de l'encodage modélisées, j'ai été chargée de générer un schéma et de documenter les décisions retenues. Ces deux étapes étaient cruciales pour le projet puisqu'elles permettaient d'abord de concrétiser nos choix d'encodage en obtenant un encodage valide et conforme au regard de l'XML-TEI, mais aussi de faciliter l'échangeabilité et la réutilisabilité de notre travail, répondant alors à notre objectif de production de données FAIR.

5.1 Le choix de l'ODD

Il est possible de documenter et de créer un schéma d'encodage de plusieurs façons. Pour le projet AGODA, nous avons décidé d'utiliser un ODD. Plusieurs raisons nous ont poussés à choisir cette solution.

5.1.1 *One Document Does it all*

Définition

Conçu et mis en œuvre par Lou Burnard et Michael Sperberg-McQueen au début du développement de la TEI [1], un ODD, acronyme pour *One Document Does it all*, est un fichier XML-TEI permettant de documenter et formaliser les choix d'encodage. Traduit de l'anglais par « un seul document fait tout », ou bien « document tout-en-un », ce dernier fournit à la fois « l'information nécessaire pour les traitements informatiques » avec le développement de spécifications du schéma d'encodage, et la « documentation de l'information destinée à être lue par un être humain » [2].

À partir de l'ODD, il est possible de générer plusieurs sorties :

— un schéma de validation reposant sur différents langages (Relax NG, XML Schema, Schematron, DTD, etc.) ;

1. *TEI : Text Encoding Initiative*, op. cit.
2. Burnard, *Qu'est-ce que la Text Encoding Initiative ?*, op. cit.

— une documentation de référence sur les éléments, les attributs, les classes d'éléments, les modèles, etc. employés dans le schéma, à l'image de celle des *Guidelines* ;
— une documentation descriptive, en prose, explicitant les choix d'encodage.

Ce document XML-TEI peut être ensuite généré sous plusieurs formats : HTML, ODT, PDF, EPUB, DOCX, etc.

Structure

Un ODD est un fichier XML-TEI, il doit donc respecter la syntaxe XML, et être conforme aux balises TEI. Il a une structure en arborescence, contenant des éléments, des attributs et des valeurs. Les balises employées sont issues du module « tagdocs », prévu spécifiquement pour la construction de la documentation et des spécifications. L'élément racine <TEI> se divise en trois sous-parties principales :

— les métadonnées du fichier intégrées dans l'élément <teiHeader> ;
— le corps de la documentation divisé en sous-parties à l'aide de l'élément et contenu dans les éléments <text> puis <body> ;
— les spécifications de l'encodage incluses dans l'élément <schemaSpec>, celui-ci contenu dans les éléments <text> puis <body>.

5.1.2 Une réponse à nos besoins

L'ODD s'est révélé être une solution en adéquation avec nos objectifs pour plusieurs raisons.

Un résultat conforme à la TEI

Tout d'abord, comme nous voulions obtenir un encodage XML-TEI conforme et valide, nous devions respecter la syntaxe XML et les directives du *framework* TEI, objectifs que nous avions mis en place lors de la modélisation. Mais nous devions également formaliser nos choix à l'aide d'un schéma spécifique à notre encodage et documenter ces derniers. L'ODD, ce document « tout-en-un » proposé par la TEI, nous permettait donc de répondre à ces deux objectifs.

Un encodage est dit « valide » s'il respecte un schéma défini. Ce dernier permet de spécifier « un ensemble de noms d'éléments, les noms et le type de données de tous les attributs qui leur sont associés, et les règles relatives aux contextes dans lesquels ils peuvent apparaître. [3] ». Un schéma correspond donc à une grammaire, déterminant l'organisation de l'encodage. Il existe plusieurs formats de schéma : Relax NG, XML Schema, Schematron, etc. La création d'un schéma nécessite de personnaliser la TEI. Celle-ci permet, en effet, de créer des spécifications propres à l'encodage mis en place. Celles-ci,

3. *Ibid.*

comme évoqué précédemment, peuvent être contenues dans l'ODD, au sein de l'élément <schemaSpec>. La personnalisation de la TEI est importante, car sans cela, l'encodage pourrait être modelé sans règle définie (mis à part le respect de la syntaxe et de l'utilisation des balises TEI) et pourrait être difficilement réutilisé par des membres extérieurs au projet. Par exemple, si nous utilisons le module « spoken » sans personnalisation, toutes les balises de ce module peuvent être appliquées. Il est donc important de contraindre un minimum les possibilités d'encodage. À partir de la définition des spécifications TEI incluses dans l'ODD, il est possible de générer un schéma, propre à ces dernières, selon différents formats. Cela consiste à convertir les spécifications XML-TEI en un langage de schéma.

Afin de faciliter l'exploitation de l'encodage ou son application par d'autres personnes, la TEI préconise de documenter les choix effectués. En effet, les contraintes sémantiques du schéma doivent être comprises par tous, afin de pouvoir être réutilisées. C'est pourquoi une documentation est essentielle. Celle-ci peut être incluse dans l'ODD au sein des spécifications, explicitant alors chacune des contraintes spécifiées, et elle peut également être développée en amont de ces dernières, sous forme de texte en prose expliquant le modèle d'encodage et justifiant les choix effectués.

Un résultat standard, pérenne et valorisable

Par ailleurs, l'ODD est un document faisant partie intégrante du système TEI. En effet, il est construit en XML-TEI (module « tagDocs ») et sa définition, sa structure et ses éléments sont développés au sein d'un chapitre qui lui est spécifiquement dédié[4]. Son fonctionnement est également développé dans le dernier chapitre des *guidelines*[5]. Ainsi, comme la TEI explique comment communiquer concrètement les choix d'encodage, l'utilisation de l'ODD nous permettait de le faire de façon standard.

De plus, comme nous avons fait le choix d'utiliser pour la documentation un format standard et que celui-ci est maintenu régulièrement à jour, cela permet d'avoir une certaine garantie sur la pérennité de notre travail. En effet, un standard peut être amené à disparaître et à être remplacé par de nouveaux. Mais grâce à la maintenabilité de celui-ci, les conversions vers de nouveaux standards pourront sûrement être effectuées, permettant alors la double conservation de l'encodage et de la documentation dans le temps.

Enfin, l'ODD peut être valorisé de différentes manières. Il peut être converti dans d'autres formats, ces derniers facilitant alors la lecture du fichier XML-TEI lui-même. Il est possible en effet de le convertir en fichier PDF, EPUB, ODT, HTML, etc. selon les besoins des projets.

4. Se référer au chapitre « Documentation Elements » des *guidelines* de la TEI, disponible sur le site web suivant : *TEI : Text Encoding Initiative*, op. cit.

5. Se référer au chapitre « Using the TEI » des *guidelines* de la TEI, disponible sur le site web suivant : *Ibid.*

5.2 La création de l'ODD

Le schéma et sa documentation[6] sont « la concrétisation du modèle conceptuel et l'ultime outil manipulé par des humains.[7] ». J'ai donc réalisé ces deux étapes, en complément de l'encodage test.

5.2.1 Génération du schéma d'encodage

Customiser la TEI

La TEI est un modèle abstrait pouvant être personnalisé selon ses besoins d'encodage, afin de créer un schéma unique propre à ce dernier. Cette customisation de la TEI prend en compte plusieurs actions : la suppression (éléments, attributs, classes entières), l'ajout (éléments au sein d'une classe, attributs ou valeurs d'attribut admis par une classe ou un élément), la modification (modèle de contenu d'un élément, usage par rapport aux *Guidelines*), le renommage d'éléments[8]. Ces spécifications permettent de construire un schéma uniforme et clair. En effet, elles constituent un ensemble de règles propres à un encodage, et contraignent les pratiques possibles lors de l'application de celui-ci.

Afin de construire un schéma propre au modèle d'encodage élaboré pour les débats parlementaires, nous avons décidé de restreindre la TEI, autrement dit nous avons souhaité définir de manière plus stricte les possibilités d'encodage, plutôt que d'étendre celles proposées initialement par la TEI. Ces dernières répondaient à nos besoins, nous n'avions donc pas besoin de nous en écarter. Les spécifications nous ont permis de préciser les balises et les attributs sélectionnés, et leur contexte d'utilisation. Nous avons également indiqué, pour certains cas, des valeurs précises d'attributs. Il était possible de pousser les spécifications plus loin, en contraignant davantage les possibilités, par exemple en rendant des enchaînements de balises obligatoires, ou en contraignant le contexte d'utilisation. Nous n'avons toutefois pas souhaité le faire, car d'une part, l'encodage devait pouvoir s'appliquer sur un grand nombre de comptes rendus, ces derniers pouvant différer sur certains points. D'autre part, comme le modèle d'encodage n'était pas fixe à ce stade du projet, nous ne pouvions pas mettre en place des spécifications poussées, ni des règles très contraignantes, puisqu'il pouvait être amené à changer.

Le choix de l'outil

Pour créer les spécifications du modèle d'encodage, nous avions plusieurs solutions à disposition. Il était possible de le faire manuellement avec un éditeur XML, dans l'ODD

6. Se référer à l'annexe B.3.

7. Calderan, Hidoine et Millet, *Métadonnées*, op. cit.

8. Albouy (Ségolène), *Les Schémas XML - Cours*, 10 janv. 2022, URL : `https://github.com/Segolene-Albouy/XML-TEI_M2TNAH/blob/main/10-ODD/2022-01-10-Introduction_ODD.pdf` (visité le 14/08/2022).

directement, au sein de l'élément <schemaSpec>. Mais cette solution longue et laborieuse, n'était pas la plus efficace, ni la plus recommandée, car nous aurions fait des erreurs. Nous pouvions utiliser, par ailleurs, *ODD by example*, qui est un scénario de transformation XSLT fourni par la TEI. Au lieu de construire le code à la main, ce dernier permet de générer automatiquement, à partir de l'encodage réalisé, un ODD avec ses spécifications. Malgré son efficacité, cette solution n'a pas été retenue, car elle ne nous permettait pas de construire le schéma progressivement en fonction des avancées de l'encodage test. Nous aurions dû regénérer, à chaque changement d'encodage, un ODD, ce qui n'était pas le plus pratique, d'autant plus que nous avions commencé la rédaction de la documentation en son sein. Nous aurions pu toutefois effectuer des modifications à la main, dans le code directement, afin de mettre à jour les spécifications générées. Mais nous avons décidé de nous tourner, finalement, vers une autre solution. Nous avons choisi d'utiliser l'outil Roma [9]. Cet outil, lui aussi proposé par la TEI, est un programme permettant de générer automatiquement un ODD, grâce à une interface web. Contrairement à *ODD by example*, les spécifications sont définies à la main, à l'aide de fonctionnalités simples proposées par l'interface, et sont ensuite récupérées, via l'interface, au format XML-TEI. Roma permet de générer également le schéma selon différents formats, ce qui n'était pas pris en compte par les autres méthodes. Nous avons trouvé que cette solution répondait au mieux à nos besoins. En effet, Roma nous permettait de prendre en compte les éléments de notre encodage progressivement, en fonction de notre avancée. Nous pouvions également modifier les spécifications grâce au système d'import de fichier proposé par l'interface. Cette solution a été retenue aussi, car elle était une pratique connue et maîtrisée par les membres de l'équipe.

L'usage de l'outil Roma [10]

Roma nous a permis de créer étape par étape les spécifications de l'encodage test de notre projet. Cet outil propose, pour commencer, plusieurs méthodes de personnalisation. J'ai choisi celle par « réduction » me permettant de personnaliser mon schéma en supprimant les modules, les éléments et les attributs du modèle entier de la TEI que je ne voulais pas conserver. J'ai ensuite renseigné quelques paramètres, correspondant aux métadonnées futures de l'ODD, à savoir le titre de la personnalisation, le titre du futur fichier, la langue, etc. J'ai ensuite procédé à la personnalisation, en m'occupant principalement de l'intégration des éléments et des attributs à garder, et en précisant parfois, quelques paramètres sur ces derniers. Pour ce faire, j'ai supprimé les modules que je ne souhaitais pas utiliser pour l'encodage, en veillant à garder les quatre modules obliga-

9. *Roma : Generating Customizations for the TEI*, URL : `https://roma.tei-c.org/` (visité le 14/08/2022).

10. Pour créer les spécifications de notre encodage, nous avons utilisé l'ancienne version de Roma *Ibid.*

toires de la TEI (modules tei, core, header et textstructure). J'ai ensuite consulté chaque module un par un, et j'ai exclu tous les éléments inutiles. En outre, j'ai désélectionné, pour chaque élément, tous les attributs que je ne voulais pas garder. Enfin, j'ai paramétré l'utilisation de certains attributs en définissant des listes fermées de valeurs, ces dernières permettant de contraindre les valeurs de l'attribut en question. J'ai attribué à chaque attribut le mode optionnel, et je n'ai pas effectué plus de contraintes, car mon objectif était de pouvoir construire des spécifications souples, au vu de l'avancée de la modélisation de l'encodage. J'ai conservé en parallèle, au sein d'un document de travail, tous les choix effectués, dans le but de garder une trace et de faciliter la reprise de la personnalisation en cas de besoin [11].

Une fois la construction des spécifications effectuée, j'ai généré le schéma sous format Relax NG (XML syntax) [12] ce qui m'a permis d'obtenir un fichier .rng contenant les règles définies de l'encodage [13]. J'ai pu également sauvegarder la personnalisation grâce à l'onglet « save customisation », en version XML-TEI, intégrée au sein d'un ODD [14], mais aussi dans un fichier XML-TEI à part [15].

Pour illustrer le résultat de cette personnalisation, voici ci-dessous le résultat XML-TEI des spécifications de l'élément <u>, mis en regard de l'interface Roma. Nous pouvons voir que les spécifications XML-TEI précisent tous les attributs exclus de l'encodage pour l'élément <u> à l'aide du mode « delete ». Les attributs autorisés ne sont pas apparents. Aucune liste fermée de valeur n'a été mise en place pour les attributs de cet élément.

11. Se référer au fichier schema_working_document.md en annexe B.1.
12. Relax NG : Langage de description de document XML, basé sur Relax (*REgular LAnguage description for XML*) et TREX (*Tree Regular Expressions for XML*).
13. Se référer au fichier agoda_schema.rng en annexe B.3.
14. Se référer au fichier agoda_odd.xml en annexe B.3.
15. Se référer au fichier agoda_schemaSpec.xml en annexe B.3.

```
<elementSpec ident="u" module="spoken" mode="change">
    <attList>
        <attDef ident="cert" mode="delete"/>
        <attDef ident="change" mode="delete"/>
        <attDef ident="copyOf" mode="delete"/>
        <attDef ident="corresp" mode="delete"/>
        <attDef ident="decls" mode="delete"/>
        <attDef ident="dur" mode="delete"/>
        <attDef ident="dur-iso" mode="delete"/>
        <attDef ident="end" mode="delete"/>
        <attDef ident="exclude" mode="delete"/>
        <attDef ident="facs" mode="delete"/>
        <attDef ident="n" mode="delete"/>
        <attDef ident="next" mode="delete"/>
        <attDef ident="notation" mode="delete"/>
        <attDef ident="prev" mode="delete"/>
        <attDef ident="rend" mode="delete"/>
        <attDef ident="rendition" mode="delete"/>
        <attDef ident="resp" mode="delete"/>
        <attDef ident="sameAs" mode="delete"/>
        <attDef ident="select" mode="delete"/>
        <attDef ident="source" mode="delete"/>
        <attDef ident="start" mode="delete"/>
        <attDef ident="style" mode="delete"/>
        <attDef ident="synch" mode="delete"/>
        <attDef ident="toWhom" mode="delete"/>
        <attDef ident="trans" mode="delete"/>
        <attDef ident="xml:base" mode="delete"/>
        <attDef ident="xml:lang" mode="delete"/>
        <attDef ident="xml:space" mode="delete"/>
    </attList>
```

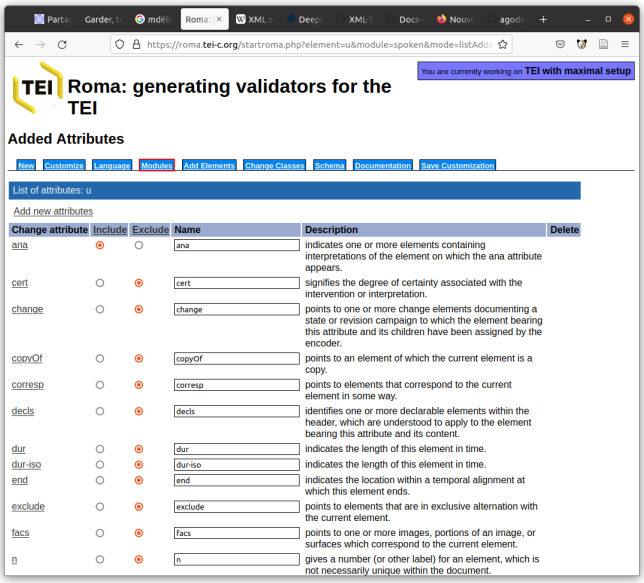


FIGURE 5.1 – Résultat XML-TEI des spécifications de l'élément <u>, mis en regard de l'interface Roma

Enfin, afin de vérifier la cohérence de ma personnalisation par rapport à l'encodage test modélisé, j'ai relié le schéma Relax NG à ce dernier en incluant dans la déclaration du fichier le nom du fichier du schéma :

```
<?xml-model href="agoda_schema.rng" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

```
<?xml-model href="agoda_schema.rng" type="application/xml"
schematypens="http://relaxng.org/ns/structure/1.0"?>
```

Lors de la validation de l'encodage, j'ai été confrontée à plusieurs erreurs de natures diverses (oublis d'éléments, oublis d'attributs, paramétrage trop restrictif). Face à cela, mon objectif n'était pas de corriger l'encodage en fonction des spécifications, mais de modifier ces dernières jusqu'à ce qu'elles correspondent à l'encodage test et permettent de le valider. J'ai donc réimporté la customisation de l'encodage dans l'interface de Roma afin de pouvoir effectuer les corrections nécessaires à l'aide des fonctionnalités de cette dernière. Mais j'ai dû gérer plusieurs problèmes de dysfonctionnement de la plateforme, qui m'ont contraint de reprendre quelques règles à la main, au sein du fichier XML-TEI. Une fois ces corrections effectuées, j'ai pu obtenir un schéma permettant de valider l'encodage test réalisé.

5.2.2 Rédaction de la documentation

Méthodologie

Après avoir généré l'ODD et obtenu les spécifications mises en place avec l'outil Roma dans l'élément <schemaSpec>, j'ai écrit une documentation en prose. Celle-ci a été rédigée à l'aide des balises TEI issues du module « tagDocs », prévues à cet effet. J'ai fait le choix d'utiliser un nombre restreint de balises, les plus essentielles, afin de faciliter la rédaction. J'ai donc employé, d'une part, les balises permettant de citer les noms de balises TEI :

— <gi> pour la mention des éléments TEI ;
— <att> pour la mention des attributs TEI ;
— <val> pour la mention des valeurs d'attributs TEI ;
— <egXML> avec l'espace de nom « `http://www.tei-c.org/ns/Examples` » qui lui est propre, pour insérer des exemples d'encodage.

D'autre part, j'ai utilisé des balises structurelles afin de hiérarchiser mes propos :

— l'élément <div1>, afin d'intégrer l'intégralité de la documentation textuelle dans une division de premier niveau ;
— les éléments <div2>, <div3> et <div4>, pour diviser la documentation en parties et sous-parties, la hiérarchie ne pouvant pas aller au-delà du quatrième niveau de division ;
— l'élément <head> afin de préciser un titre à la division en question ;
— l'élément <p> pour indiquer les paragraphes ;
— l'élément <list> contenant des <item> pour lister différentes idées.

L'encodage de la prose textuelle permet, outre la structuration et l'enrichissement des données, d'obtenir *in fine* une visualisation avec une mise en page succincte (table des matières, parties avec numérotation, titres, objets TEI et exemples mis en évidence, listes à puce, etc.). Elle permet ainsi de simplifier la lecture.

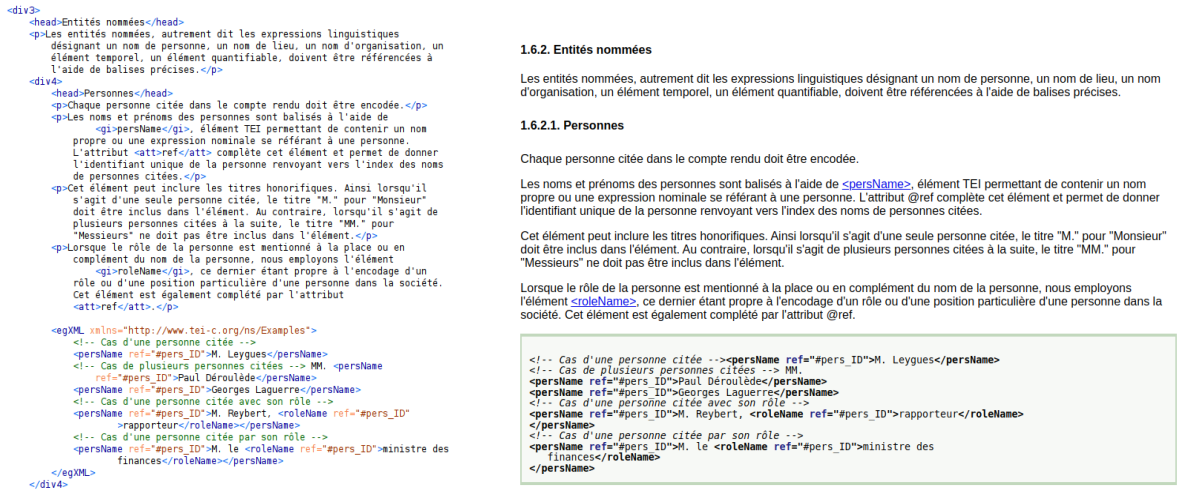


FIGURE 5.2 – Résultat des vues XML et HTML de l'ODD

Résultat

Le résultat de notre documentation en prose permet d'apporter tous les éléments de compréhension nécessaires à notre encodage. Elle informe les objectifs visés par l'encodage, elle rappelle la structure propre au document XML-TEI et explique le rôle de ses composants. Elle met en avant les balises utilisées et explicite leur contexte d'utilisation selon les cas spécifiques de notre corpus. Elle illustre les propos à l'aide de nombreux exemples.

La rédaction de la documentation a été effectuée sur les dernières semaines de mon stage, puisque les choix d'encodage ont évolué et ont été fixés que tardivement. Le plan de cette dernière a pu être pensé en amont, comme nous avions déjà les grandes idées d'encodage. Pour ce faire, je me suis inspirée de la documentation du projet ParlaMint. En effet, cette dernière étant structurée, précise et claire, je me suis appuyée dessus pour penser le plan de notre propre documentation. J'ai donc mis en place sept parties afin de traiter l'ensemble des aspects de l'encodage. La première partie apporte des propos contextuels sur le projet (éléments historiques, objectifs scientifiques, choix de transcription). Les deuxième et troisième parties contiennent des informations générales sur l'encodage (définitions de la structure du corpus et du composant, précisions sur les conventions de nommage des fichiers, informations sur les caractères, valeurs standards et taxonomies). Les trois parties suivantes portent spécifiquement sur les choix d'encodage des textes. Ces derniers sont répartis selon leur fonction. J'ai distingué pour cela les métadonnées, le balisage physique et le balisage sémantique. La dernière partie correspond à la bibliographie de la documentation. Ce plan a été approuvé par l'équipe et nous nous sommes réparti la rédaction afin de pouvoir la rédiger au plus vite. J'ai été chargée de rédiger les quatre parties suivantes : organisation générale, prérequis généraux, balisage physique et balisage sémantique. La rédaction a dû être mise à jour un certain nombre de fois, à cause des modifications tardives effectuées sur l'encodage. À l'issue de mon stage, la documentation était presque entièrement rédigée. Quelques parties restaient toutefois à traiter ou à être complétées, notamment sur des questions d'encodage mis de côté jusqu'alors.

Afin de visualiser le résultat presque final de cette dernière, nous avons souhaité générer une sortie HTML de l'ODD. Nous avons décidé de la valoriser en la stockant sur le dépôt GitHub du projet. Celui-ci étant public, l'utilisateur peut alors récupérer le document HTML en le téléchargeant sur son ordinateur, et l'ouvrir ensuite avec un navigateur. Afin de simplifier cette démarche, nous avons souhaité également le rendre accessible directement sur une page web à l'aide de Github Page [16]. Ce dernier est une option proposée par le dépôt Github afin d'héberger des sites web statiques (HTML, CSS, Javascript). Le stockage de notre fichier HTML sur ce dépôt permet ainsi de visualiser le contenu directement [17].

16. Pour plus d'informations, se référer au site web suivant : `https://pages.github.com/`.
17. Lebreton, Puren et Vernus, *AGODA*, op. cit.

Troisième partie

Appliquer l'encodage : le défi du balisage automatique

La seconde phase de la deuxième étape de la chaîne de traitement du projet AGODA consiste à appliquer l'encodage modélisé. En effet, une fois l'encodage pensé, validé et formalisé, nous devions intégrer les balises retenues dans les textes océrisés du corpus. Cependant, l'ensemble de ces textes représentaient un groupement colossal à traiter, comprenant une législature entière, soit tous les comptes rendus ayant eu lieu sur cette période de quatre ans. Aussi, ces derniers étaient, pour certains, très longs, pouvant se développer sur de nombreuses pages du journal. Il y avait, par conséquent, 10418 images numérisées à prendre en compte, ce qui était un véritable défi d'encodage. Face à cette difficulté, il nous était impossible de baliser l'ensemble de ces textes à la main pour des raisons pratiques (peu de temps, peu de main-d'œuvre). Par ailleurs, le projet AGODA avait pour but d'appliquer ce balisage sur un corpus encore plus large (1881-1940). En fonction de ces contraintes et des objectifs du projet, l'équipe a envisagé d'emblée l'automatisation du balisage. Cette méthode repose sur l'intervention d'une machine, qui a pour mission de traiter les tâches configurées par l'homme. « Automatique » correspond, en effet, à une :

> science visant l'emploi d'une machine où l'intervention humaine est limitée à la préparation préalable, intellectuelle et matérielle, d'un programme incorporé à la machine qui le suivra seule, en le modifiant d'elle-même s'il y a lieu, par des décisions logiques conditionnées par les circonstances de déroulement des opérations. [18]

L'automatisation était donc une solution pour réaliser l'encodage sur l'ensemble du corpus, sans que nous ayons besoin d'appliquer nous-mêmes les balises. Nous avions pour rôle, alors, de configurer un programme informatique, c'est-à-dire une liste d'instructions écrites sous une forme conventionnelle, permettant de gérer l'encodage, et de le corriger en fonction du résultat obtenu par l'ordinateur. J'ai été chargée de cette mission durant mon stage et il va s'agir d'expliquer et d'évaluer, dans cette partie, le processus d'automatisation que j'ai mis en place, en fonction des particularités de la source et des contraintes techniques.

18. Définition issue du CNRTL : `https://www.cnrtl.fr/definition/automatique`.

Chapitre 6

Modéliser le processus d'automatisation

Tout comme la construction du modèle d'encodage, la conception du processus de balisage automatique nécessite une phase de modélisation afin de définir l'objectif et les étapes de ce dernier, mais aussi de déterminer les besoins et de choisir les outils techniques pour le mettre en œuvre. À la suite de ces réflexions, les données ont été préparées en fonction des paramètres définis.

6.1 Définir le processus d'automatisation

La conception du processus d'encodage automatique a suscité de nombreux questionnements en amont de son application, notamment sur l'objectif et les étapes de ce dernier.

6.1.1 L' objectif initial

Tout d'abord, j'ai défini l'objectif du processus d'encodage automatique afin de fixer un objectif à atteindre. L'objectif de ce processus était d'obtenir l'ensemble des débats parlementaires du corpus encodés en XML-TEI. Chacun des numéros du *Journal officiel* devait correspondre à un fichier XML composant, celui-ci devant être relié au fichier XML corpus. Ce but initial représentait toutefois un objectif complexe à mettre en place, puisqu'il devait prendre un certain nombre de paramètres en compte. En effet, le processus d'encodage automatique reposait sur différentes contraintes liées :

— au format des débats océrisés, et au contenu de ces fichiers ;
— au nombre de fichiers à traiter ;
— à la diversité des données contenues dans ces fichiers ;
— au modèle d'encodage élaboré.

Il fallait donc pouvoir transformer les débats océrisés en plusieurs fichiers contenant les données encodées en XML-TEI, cet encodage s'appliquant en fonction des particularités des textes, et selon le modèle prédéfini.

Ces différentes contraintes ont soulevé plusieurs questionnements :

— Je me suis interrogée d'abord sur les différentes étapes à mettre en place pour atteindre l'objectif initial souhaité. La difficulté reposait sur la procédure à élaborer en fonction des besoins, et en fonction des outils et des sources à disposition ;

— Le corpus à traiter étant vaste et dense, je me suis demandé en parallèle s'il était possible d'automatiser toutes les tâches à effectuer pour obtenir les débats encodés en XML-TEI. Était-il possible d'appliquer automatiquement toutes les balises du modèle d'encodage en fonction des moyens techniques à disposition ? Était-il possible de gérer la création des fichiers composants et de les relier avec le fichier corpus ? La difficulté était donc de savoir jusqu'où pousser l'automatisation.

L'ensemble de ces questionnements a été évalué et traité tout au long de la modélisation, en gardant à l'esprit l'objectif principal du processus : créer un corpus encodé en XML-TEI selon le modèle d'encodage élaboré.

6.1.2 Du JSON au XML-TEI : transformer le format des données

Le format des données océrisées

Le processus d'encodage automatique a été pensé en fonction du format initial des données. Un format de données correspond à la façon dont est représenté un type de données ; c'est une convention structurelle, permettant d'être comprise par les programmes informatiques et logiciels. Suite à l'océrisation des images numérisées des débats parlementaires réalisée à l'aide de l'outil d'OCR du LRDE, nous avons obtenu des données textuelles manipulables au format JSON [1]. La sortie de l'OCR était en effet configurée pour être sous ce format. Le JSON est un format structurant les données qui le compose sous forme de couples clé et valeur, inclus au sein d'une liste de valeurs ordonnées.

Par ailleurs, le LRDE a mentionné la possibilité de récupérer l'océrisation des textes au format texte brut à l'aide du sed [2] JQ. Ce dernier permet d'extraire le texte océrisé du flux JSON, et d'obtenir les données en texte brut, constituées uniquement de caractères simples, sans structuration particulière. Afin d'obtenir cette sortie, nous devions effec-

1. Acronyme de *JavaScript Object Notation*.

2. Acronyme de Streem EDitor, sed est un « programme informatique permettant d'appliquer différentes transformations prédéfinies à un flux séquentiel de données textuelles. » Définition issue du site web suivant : `https://fr.wikipedia.org/wiki/Stream_Editor`.

tuer dans l'interface système[3] de l'ordinateur plusieurs commandes, ce qui était plutôt contraignant.

Face à ces deux formats, nous avons souhaité utiliser la sortie JSON des textes océrisés. Basé sur un sous-ensemble du langage de programmation JavaScript, ce format est pensé et conçu pour l'échange des données. En effet, il propose des structures de données universelles, partagées pratiquement par tous les langages de programmation. Il est donc un format interopérable, facilement échangeable, et convertissable. Il présentait alors un avantage considérable pour notre projet, puisque nous pouvions l'utiliser comme format pivot, celui-ci pouvant être transformé selon nos besoins.

Convertir le format des données

J'ai donc d'abord défini le processus d'automatisation de l'encodage comme une conversion de format.

FIGURE 6.1 – Schéma illustrant la conversion automatique

Notre objectif était, en effet, d'obtenir, à partir des données au format JSON, des données au format XML-TEI. Nous devions donc mettre en place une conversion du format initial des données, en transformant la structure JSON vers la structure XML. Cette conversion avait pour contrainte de respecter le modèle d'encodage XML-TEI élaboré.

6.1.3 Créer une chaîne de traitement d'automatisation

De plus, le processus d'automatisation de l'encodage ne repose pas que sur le seul objectif de conversion des données. En effet, le but est d'obtenir des fichiers composants XML-TEI conforme à la TEI, reliés à un fichier XML corpus, et valides auprès du schéma défini. Cela nécessite alors la mise en place d'une chaîne de traitement spécifique pour gérer automatiquement, à la fois la conversion des données, et les opérations de stockage et de validation. Afin de comprendre les différentes opérations à appliquer, j'ai réalisé un schéma synthétique mettant en avant les différents besoins pour constituer le résultat souhaité. Ce schéma relate alors les grandes étapes de la chaîne de traitement :

3. Interface système, ou en anglais *shell*, est « un programme qui reçoit des commandes informatiques données par un utilisateur à partir de son clavier pour les envoyer au système d'exploitation qui se chargera de les exécuter. » Définition issue du site web suivant : `https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445272-shell-en-informatique-definition-simple-role-et-exemples/`.

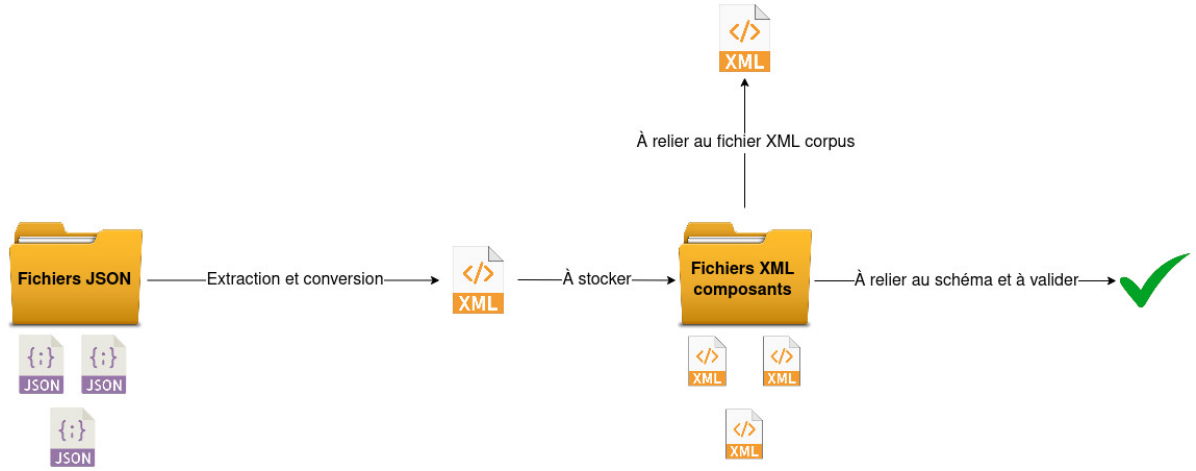


FIGURE 6.2 – Schéma de la chaîne de traitement d'automatisation

La première étape consiste à extraire les données textuelles du fichier JSON afin de les encoder en y ajoutant les différentes balises, en vue de la création du fichier XML composant, celui-ci comportant les données relatives à un seul numéro de journal. La deuxième étape consiste à stocker le résultat dans un dossier contenant l'ensemble des fichiers XML composants, et à le relier au fichier XML corpus par le mécanisme d'inclusion XInclude. Enfin, la troisième étape repose sur la validation des fichiers XML-TEI obtenus.

6.2 Réflexion méthodologique

L'élaboration du processus d'automatisation a nécessité une phase réflexive qui s'est déroulée en trois étapes principales : j'ai d'abord analysé les contenus des fichiers JSON afin de comprendre leur structure et de mieux appréhender leur conversion, j'ai ensuite choisi le langage de programmation et conceptualisé la méthodologie technique à mettre en place. Enfin, j'ai aidé Marie Puren pour l'élaboration du guide d'annotation, celui-ci constituant le principal outil d'aide à la conversion des données.

6.2.1 Analyser les fichiers JSON

La conversion d'un format nécessite une bonne connaissance du contenu et de la structure du format de départ, et du format d'arrivée. Le modèle d'encodage XML-TEI que nous souhaitons atteindre a fait l'objet d'une analyse développée dans la partie précédente. Nous allons donc analyser, à présent, les fichiers de départ à convertir, c'est-à-dire les fichiers JSON contenant le texte océrisé.

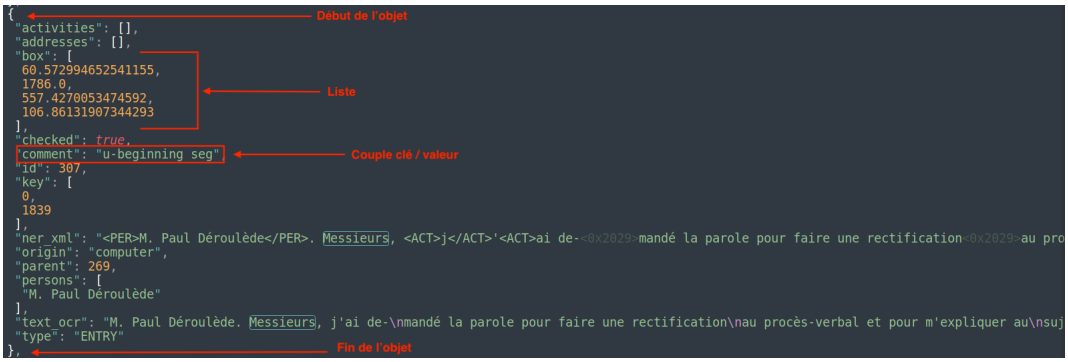


FIGURE 6.3 – Illustration de la structure d'un JSON

Le contenu des fichiers JSON est structuré selon les normes de ce format. Ces fichiers sont donc composés d'un « tableau », autrement dit d'une liste de valeurs ordonnées. Cette liste s'ouvre et se ferme à l'aide de crochets. Les valeurs de cette liste sont des objets, appelés aussi dictionnaires, et sont contenues entre accolades. Enfin, les dictionnaires sont constitués d'un ensemble de couples clé / valeur non ordonné. Chaque nom de clé est suivi de deux-points, et chaque couple clé / valeur est séparé par une virgule. Les valeurs associées aux clés peuvent être de plusieurs types : chaîne de caractères, nombre, true ou false ou null, objet, ou encore tableau [4].

Le contenu des fichiers JSON correspond au résultat de l'OCR et est donc présenté selon la structure particulière du format que nous venons d'exposer. Il est constitué de trois informations principales correspondant à trois des étapes du processus d'océrisation de l'outil du LRDE : la segmentation, la reconnaissance des caractères, l'enrichissement des données (reconnaissance des entités nommées, annotation).

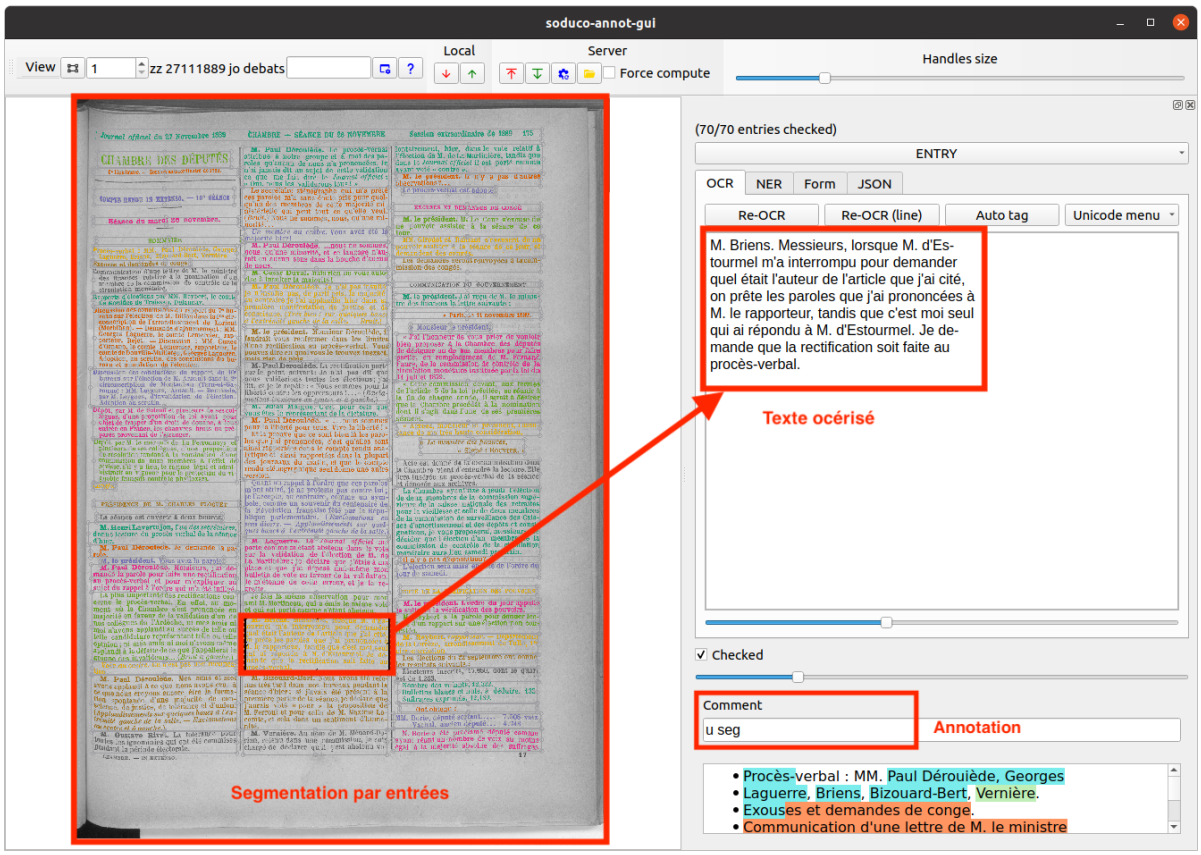
Chaque objet contenu dans la liste initiale du fichier JSON constitue d'abord une zone segmentée de l'image numérisée. Ces objets contiennent en effet les coordonnées de cette dernière, mentionnées dans une liste de valeurs et rattachées à la clé « box ». Le niveau de segmentation (page, section, colonne, paragraphe, ligne) est indiqué grâce à la clé « type ». Cette même clé peut également contenir une information de type structurel (titre de niveau 1, titre de niveau 2). Un identifiant unique est aussi associé à chacune des zones afin de pouvoir indiquer leur hiérarchie. Une zone pouvant contenir en son sein une autre zone, l'identifiant unique permet alors de mentionner l'identifiant de la zone ainsi que l'identifiant parent de ce dernier. (ex : une colonne contient les paragraphes, les paragraphes associés à cette colonne partagent donc le même identifiant unique parent). De plus, un couple clé / valeur permet d'indiquer si la segmentation a été validée (clé « *checked* », valeur « *true* » ou « *false* »). En outre, un fichier JSON correspond à la segmentation d'une seule image numérisée. La liste de valeurs initiale de ce dernier contient alors les informations d'une seule page de compte rendu. Cela signifie par exemple que, le

4. Analyse des éléments structurels issus du site web suivant : `https://www.json.org/json-fr.html`.

compte rendu de la séance du 26 novembre 1889, constitué de 19 images numérisées, est divisé en 19 fichiers JSON.

Outre les informations liées à la segmentation, chaque zone segmentée, autrement dit chaque objet de la liste de valeurs ordonnées, contient les données extraites de la zone. Le texte océrisé est situé dans les objets de type « ENTRY », c'est-à-dire dans les zones segmentées de niveau paragraphe, et correspond à la valeur associée à la clé « text_ocr ». Il peut être situé également dans les objets de type « TITLE_LEVEL_1 » et « TITLE_LEVEL_2 », correspondant au texte issu des zones particulières telles que le bandeau ou la manchette.

Enfin, chaque objet contient des couples clé / valeur apportant des informations ajoutées à la zone océrisée. En effet, l'outil du LRDE permet la reconnaissance des entités nommées du texte, et l'annotation manuelle des zones de paragraphes à l'aide de tags. Ces informations sont donc incluses au sein du fichier JSON. La clé « ner_xml » a pour valeur le texte océrisé enrichi avec le balisage des entités nommées. Les balises <PER>, <ACT>, <LOC>, <TITRE>, <CARDINAL> permettent respectivement d'annoter les noms et rôles des personnes, les noms de lieux, les noms d'organisations, et les nombres. La clé « comment » a pour valeur le ou les tags associés à la zone du paragraphe.



(a) Interface de l'outil d'OCR


```
{
 "activities": [],
 "addresses": [],
 "box": [
  651.7002410410179,
  1988.0,
  550,
  208
 ],
 "checked": true,
 "comment": "u seg",
 "id": 343,
 "key": [
  1,
  2092
 ],
 "ner_xml": "<PER>M. Briens</PER>. Messieurs, lorsque M. <PER>d</PER>'<PER>Es</PER><ACT>-</ACT><PER><0x2029>tourmel</PER> m'a interrompu <ACT>
    pour demander<0x2029>quel était l</ACT>'<ACT>auteur de l</ACT>'<ACT>article que j</ACT>'ai cité,<0x2029>on prête les paroles que j'ai
    prononcées à<0x2029>M. le rapporteur, tandis que c'est moi seul<0x2029>qui ai répondu à M. d'Estourmel. Je de-<0x2029>mande que la
    rectification soit faite au<0x2029>procès-verbal.",
 "origin": "computer",
 "parent": 272,
 "persons": [
  "M. Briens",
  "d",
  "Es",
  "tourmel"
 ],
 "text_ocr": "M. Briens. Messieurs, lorsque M. d'Es-\ntourmel m'a interrompu pour demander\nquel était l'auteur de l'article que j'ai cité,\non
    prête les paroles que j'ai prononcées à\nM. le rapporteur, tandis que c'est moi seul\nqui ai répondu à M. d'Estourmel. Je de-\nmande que
    la rectification soit faite au\nprocès-verbal.",
 "type": "ENTRY"
},
```


(b) Sortie JSON obtenue

FIGURE 6.4 – Comparaison entre l'océrisation d'un extrait de texte et la sortie JSON obtenue

6.2.2 Choix techniques

Questionnements préliminaires

Je me suis questionnée, en parallèle de cette analyse, sur les différentes possibilités techniques pour traiter les fichiers JSON selon les étapes de la chaîne de traitement. Pour que la conversion du format et la gestion du résultat obtenu soient automatisées, je devais réfléchir à la méthodologie technique et au(x) langage(s) de programmation à utiliser. Je me suis d'abord interrogée sur la conversion du format de données, en me demandant s'il valait mieux convertir directement les fichiers JSON en fichiers XML-TEI ; ou s'il était préférable de diviser les étapes en passant d'abord du JSON au XML, puis en transformant dans un deuxième temps le XML en XML-TEI. De plus, je me suis questionnée sur l'utilité des composants des fichiers JSON et si je devais, par conséquent, conserver ou non l'ensemble de ces composants lors de la conversion du format. En outre, comme la chaîne de traitement du processus d'automatisation reposait à la fois sur la conversion des données, mais aussi sur la gestion des fichiers XML-TEI obtenus, je me suis interrogée sur le nombre de langages de programmation nécessaire pour gérer l'ensemble de ces tâches différentes. Ces questions ont pu être résolues lors de l'analyse des avantages et des inconvénients des différents langages de programmation.

Réflexion sur la méthode en fonction des langages à disposition

Un langage de programmation est un « système d'instructions et de règles syntaxiques servant à la programmation informatique [5] ». Il permet de construire des instructions, lisibles par l'ordinateur, et contribuant à l'automatisation des tâches. Le format JSON étant un format pivot qui peut être traité par n'importe quel langage de programmation, j'ai décidé d'axer mon analyse sur deux langages que j'ai appris à utiliser durant mon année de Master 2 : XSLT et Python.

L'*Extensible Stylesheet Language Transformations* (XSLT) est un « langage de transformation qui permet de transformer l'arbre XML d'un document en un autre arbre XML ou un autre format textuel. [6] ». Ce langage pouvait être un moyen de traiter la conversion des données, en transformant la structure JSON en un arbre XML et selon les normes TEI. L'éditeur XML Oxygen que nous utilisions mettait à disposition une feuille de style XSL contenant un ensemble d'instructions déjà définies prenant en charge la transformation du format JSON au format XML [7]. Cette méthode pouvait nous permettre alors de convertir le format des données en deux temps, en passant du JSON au XML à l'aide

5. Définition de langage de programmation issue du dictionnaire La langue française, disponible à l'adresse suivante : `https://www.lalanguefrancaise.com/dictionnaire/definition/langage-de-programmation#0`.

6. Calderan, Hidoine et Millet, *Métadonnées*, op. cit.

7. Méthode d'application de la feuille XSL d'Oxygen disponible sur le lien suivant : `https://www.oxygenxml.com/doc/versions/24.1/ug-editor/topics/convert-JSON-to-XML-x-tools.html`.

de cette feuille de style prédéfinie, puis en transformant à l'aide d'une feuille de style XSL maison les données XML vers notre modèle d'encodage XML-TEI. Cependant, cette première solution n'était pas optimale pour plusieurs raisons. D'abord, je n'ai pas trouvé le contenu de la feuille de style XSL implémentée dans l'éditeur Oxygen. Ce dernier ne mettait à disposition que la fonctionnalité permettant d'appliquer la conversion. Je n'ai donc pas pu analyser le code de la feuille de transformation et comprendre son fonctionnement. De plus, cette feuille de style XSL convertissait l'ensemble des objets du JSON dans un arbre XML, transformant chacune des clés et des valeurs en élément XML, avec pour nom de balise la clé et pour valeur le contenu de la clé. L'agencement de ces derniers dépendait de la structure hiérarchique des objets JSON.

Cependant, notre modèle d'encodage en XML-TEI était très éloigné de ce premier résultat et nous contraignait à construire une feuille de style complexe, prenant en compte l'extraction des nœuds XML de l'arbre utiles pour notre modèle, puis l'application des balises TEI spécifiques. En outre, cette méthode était aussi d'un usage limité. En effet, au vu de la grandeur de notre corpus, la conversion permise par la fonctionnalité d'Oxygen n'était pas idéale puisqu'elle devait être appliquée manuellement à l'aide de cet éditeur. Cet inconvénient nous rendait par la même dépendants du logiciel propriétaire Oxygen, et limitait la reproductibilité de la *pipeline*. Enfin, ce langage de transformation ne nous permettait pas de prendre en compte l'intégralité des besoins de la chaîne de traitement d'automatisation, notamment sur la gestion des fichiers résultats obtenus suite à la conversion. Nous devions donc envisager d'utiliser en parallèle un autre langage de programmation.

Cette première analyse m'a conduite à me tourner vers le langage de programmation Python. Ce dernier peut s'adapter à tout type de besoins grâce à ces nombreuses bibliothèques spécialisées [8] et est très utilisé notamment pour l'automatisation des tâches. Il permettait de traiter la conversion du format JSON, mais aussi de répondre à nos autres besoins de la chaîne de traitement. Python nous proposait pour cela différents outils tels que des librairies spécifiques pour la gestion de nos deux formats, et nous permettait, grâce à sa syntaxe, de mettre en place différentes tâches d'automatisation. Lors de l'analyse des différentes librairies de ce langage, je me suis intéressée d'abord à la librairie *jsontoxml* puisqu'elle permettait de convertir le format JSON vers le format XML. Mais je suis arrivée au même constat que pour XSLT : ce type de conversion est compliqué à mettre en place pour notre corpus, puisqu'il nécessite de convertir les données en deux temps. J'ai donc réaxé mes recherches sur d'autres librairies me permettant de traiter les fichiers JSON autrement.

8. Une bibliothèque logicielle est un ensemble de modules, autrement dit « un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire ». Définition issue du site web suivant : `https://www.techno-science.net/definition/1470.html`.

Définition de la méthode envisagée avec Python

J'ai décidé de privilégier une conversion des données en une seule étape afin d'être plus efficace et d'obtenir directement le modèle XML-TEI souhaité. Cette solution était plus rapide et plus facile à maintenir qu'un protocole en plusieurs étapes. Je devais donc penser des scripts « maison » afin d'adapter la transformation à nos besoins spécifiques. Pour cela, j'ai fait le choix d'employer le package « json » [9], celui-ci permettant de parser les fichiers JSON à l'aide de la méthode .load(). Je pouvais grâce à cette méthode extraire les objets utiles à mon encodage futur pour ensuite les convertir à l'aide de règles de transformations spécifiques. Par ailleurs, la librairie *lxml*, pensée pour le traitement XML en python, me permettait de gérer les autres tâches de la chaîne de traitement, à savoir l'inclusion des fichiers composants dans le fichier XML corpus, et la validation des fichiers XML au regard du schéma.

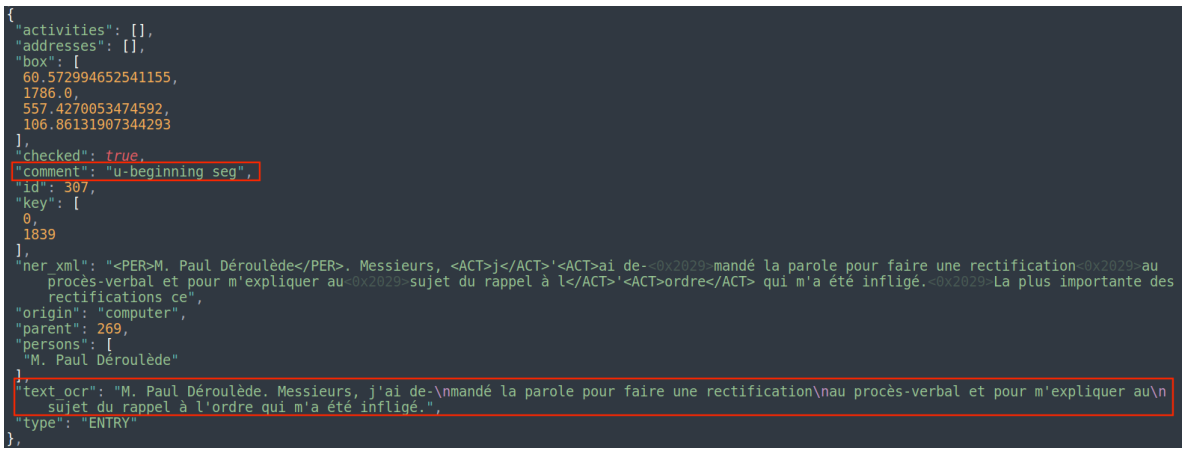
6.2.3 Élaborer un guide d'annotation

Afin de convertir les données en XML-TEI, nous avons eu besoin d'extraire des fichiers JSON les éléments utiles pour l'encodage. Il était essentiel de récupérer d'abord le contenu textuel des comptes rendus. Ces données océrisées correspondaient aux valeurs des clés « text_ocr ». Celles-ci étaient intégrées dans les objets correspondant à la segmentation de niveau paragraphe, divisant ainsi le texte selon sa structure en paragraphe. L'objectif était de pouvoir ensuite appliquer des règles de transformations sur les valeurs de ces clés « text_ocr », dans le but de baliser le texte selon le modèle d'encodage défini. Pour poser l'ensemble des balises de cet encodage, nous avions besoin de trouver des *features*, autrement dit des points de repère dans le texte, pouvant être de nature structurelle ou typographique. Par exemple, si nous souhaitions encoder les interventions des orateurs à l'aide de l'élément <u>, nous devions pouvoir trouver un trait caractéristique au début et à la fin de l'intervention pour pouvoir situer les balises ouvrante et fermante de l'élément <u>. Les prises de paroles étant toujours introduites par l'indication en gras du nom et/ou du titre de l'orateur suivi d'un point, cela pouvait constituer un repère textuel utile pour la pose des balises. Cependant, bien que les comptes rendus aient une construction homogène, ils restent complexes à traiter, car ils ne présentent que trop peu de *features* textuels récurrents. L'encodage modélisé prenant en compte un grand nombre d'éléments différents, de nature logique, formelle et sémantique, ils nous étaient impossible de transformer tous les éléments puisque nous n'avions pas de point de repère. Il était, par exemple, impossible de structurer les différentes parties des comptes rendus à l'aide de l'élément , car nous ne pouvions pas nous appuyer sur une structure ou un

9. *Json — JSON Encoder and Decoder — Python 3.10.6 Documentation*, URL : `https://docs.python.org/3/library/json.html` (visité le 14/08/2022).

élément formel particulier. En effet, elles n'étaient pas toujours introduites par un titre, et ne commençaient ni ne finissaient avec un mot ou un signe récurrent.

Face à ce constat, nous avons cherché dans les fichiers JSON des couples de clé / valeur pouvant nous aider dans la construction des règles de transformation. Nous nous sommes intéressés à la clé « ner_xml » ayant pour valeur le texte océrisé enrichi incluant la reconnaissance des entités nommées. Le texte était alors balisé selon les étiquettes définies par l'outil d'OCR du LRDE (<PER>, <LOC>, <ACT>, etc.). Ce couple de clé / valeur était alors un élément important pouvant nous aider à appliquer l'encodage des entités nommées. Nous pouvions extraire le contenu de cette clé, et transformer simplement le nom des balises selon la norme TEI. Par exemple, l'étiquette <PER> encodant les noms de personne pouvait être échangée avec la balise <persName> propre à la TEI. De plus, nous avons relevé la clé « comment » prenant pour valeur un ou des tag(s). Cette dernière était le résultat de l'annotation effectuée au sein de l'interface de l'outil d'OCR du LRDE. L'annotation permettait en effet d'associer des étiquettes aux différentes zones segmentées et représentait un enrichissement des données océrisées. Elle était pour nous un apport considérable, puisque les tags pouvaient constituer des points de repère pour appliquer les règles de transformation sur chacune des zones textuelles de niveau paragraphe. Par exemple, si la zone segmentée correspondait à une prise de parole, nous pouvions l'indiquer à l'aide d'un tag particulier, puis, à l'aide de ce dernier, appliquer la règle de transformation spécifique à ce cas, ayant pour rôle d'ajouter dans le texte l'élément <u> prévu pour l'encodage de ce type d'information. Cette méthode nous permettait de pouvoir prendre en charge tous les éléments de l'encodage, qu'ils soient de nature formelle, logique ou sémantique, à l'exception des entités nommées que nous pouvions traiter avec la clé « ner_xml » évoquée précédemment. Nous avons donc retenu cette solution et, par souci d'homogénéisation et de simplification de la conversion, nous avons décidé de construire nos règles de transformation qu'à partir de ces tags, laissant alors de côté certains composants du JSON.



```
{
 "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
    procès-verbal et pour m'expliquer au<0x2029>sujet du rappel à l</ACT>'<ACT>ordre</ACT> qui m'a été infligé.<0x2029>La plus importante des
    rectifications ce",
 "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\n
    sujet du rappel à l'ordre qui m'a été infligé.",
 "type": "ENTRY"
},
```


FIGURE 6.5 – Couples clé / valeur à extraire des fichiers JSON

La mise en place de cette méthode a nécessité l'élaboration d'un guide d'annotation [10]. Comme son nom l'indique, il avait pour but de guider l'annotation en mettant à disposition une liste de tags et en définissant leur contexte d'utilisation afin qu'ils puissent être utilisés selon une norme commune. Nous avons donc défini des tags en fonction des spécificités de l'encodage. Le guide d'annotation indique le nom des étiquettes, leur contexte d'utilisation, et la modélisation du résultat XML-TEI de la conversion de la zone étiquetée. Puisque chaque tag correspondait à un morceau d'encodage XML-TEI, les règles de transformation devaient se baser sur ces indications.

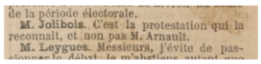
Tags	Utilisation	Balises TEI	Exemples
u seg	Prise de parole correspondant à un seul paragraphe.	<u> <seg>text</seg> </u>	

FIGURE 6.6 – Extrait du guide d'annotation pour l'étiquette u

Comme illustré ci-dessus, le tag « u » couplé avec le tag « seg » permettent d'encoder une prise de parole contenue sur un seul paragraphe, et sont rattachés au fragment du modèle d'encodage spécifique à cette information.

Cependant, la construction du guide a été très compliquée pour plusieurs raisons. D'une part, nous devions prendre en compte toutes les caractéristiques de l'encodage visé, tout en s'assurant que l'assemblage des fragments du modèle d'encodage défini par les différents tags forme bien un document XML-TEI conforme et valide. En effet, nous ne devions pas nous tromper sur la construction des fragments d'encodage, puisque, sinon, nous risquions d'avoir des enchevêtrements de balises, ou des éléments non fermés. D'autre part, les particularités de la structure du JSON nous ont contraints de complexifier notre guide d'annotation. Le texte étant divisé selon la segmentation de l'océrisation, nous devions annoter les zones relatives à un niveau de segmentation, et nous avons choisi d'annoter sur le niveau de type *entry*, c'est-à-dire de niveau paragraphe. Or certains éléments pouvaient s'étendre sur plusieurs paragraphes, sur plusieurs colonnes et même sur plusieurs pages. Ces éléments étaient alors divisés au sein des valeurs de plusieurs clés « text_ocr » des fichiers JSON. Il fallait trouver une solution pour taguer le début de l'information et la fin de cette dernière, afin que l'information soit bien balisée avec un élément ouvrant et un élément fermant. Prenons l'exemple d'une prise de parole. Celle-ci pouvait être développée sur plusieurs paragraphes, voire sur plusieurs colonnes et sur plusieurs pages.

10. Se référer au fichier guide_annotations_agoda.pdf en annexe C.1.

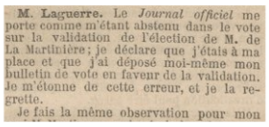
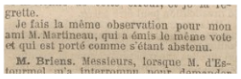
u-beginning seg	Premier paragraphe d'une prise de parole qui se poursuit sur un paragraphe ou plus	<u> <seg>text</seg>	 **M. Laguerre.** Le *Journal officiel* me porte comme m'étant abstenu dans le vote sur la validation de l'élection de M. de La Martinière; je déclare que j'étais à ma place et que j'ai déposé moi-même mon bulletin de vote en faveur de la validation. Je m'étonne de cette erreur, et je la regrette. Je fais la même observation pour mon
u-end seg	Dernier paragraphe d'une prise de parole qui a débuté avec un ou plusieurs paragraphes	<seg>text</seg> </u>	 grette. Je fais la même observation pour mon ami M. Martineau, qui a émis le même vote et qui est porté comme s'étant abstenu. **M. Briens.** Messieurs, lorsque M. d'Es-

FIGURE 6.7 – Extrait du guide d'annotation pour les étiquettes u-beginning et u-end

Pour ces cas, nous avons décidé de définir deux tags supplémentaires, « u-beginning » et « u-end », ces derniers devant être annotés sur le premier paragraphe de l'intervention, puis sur le dernier, afin de pouvoir obtenir ensuite les balises ouvrante et fermante de l'élément <u>, placées aux bons endroits. Cette contrainte a concerné un grand nombre des tags du guide d'annotation. Enfin, la construction du guide d'annotation s'est avérée difficile, car notre modèle d'encodage a évolué jusque tardivement. Nous devions alors mettre à jour les étiquettes dès que l'encodage évoluait.

6.3 Préparation des données

Une fois le processus d'encodage automatique défini et la méthodologie approuvée, j'ai effectué, avec l'aide de l'équipe, un travail préparatoire avant d'appliquer l'encodage. Pour cela, nous avons sélectionné d'abord les données à utiliser, puis nous les avons préparées pour le futur traitement automatique.

6.3.1 Choix des données tests

Afin de tester le processus d'automatisation, nous avons décidé de ne travailler qu'avec deux comptes rendus seulement, celui de la séance du 26 novembre 1889 et celui de la séance du 14 janvier 1890. Le choix de ces deux textes était simple : d'une part ils étaient représentatifs d'un maximum de cas d'encodage à traiter, ce qui nous permettait alors de tester une grande partie des conversions prévues à partir des étiquettes. La première séance, utilisée pour l'encodage test, représentait un compte rendu type, contenant les particularités textuelles les plus répandues sur l'ensemble du corpus. La deuxième séance, quant à elle, nous permettait de traiter le cas particulier des numéros contenant plusieurs comptes rendus. D'autre part, comme nous avions réalisé le modèle d'encodage sur la séance du 26 novembre 1889, cela nous donnait la possibilité de comparer facilement le modèle d'encodage visé, de l'encodage obtenu après l'application du processus d'automatisation, ce qui représentait une aide précieuse. En outre, il était important de prendre en compte au moins deux comptes rendus afin de tester certaines fonctionnalités de la chaîne de traitement, liées à la gestion de l'organisation des fichiers.

Ainsi, ces deux séances sélectionnées représentaient pour nous des données tests, riches et fonctionnelles, nous permettant de faire des essais et d'évaluer ensuite le résultat pour effectuer les ajustements nécessaires dans le code informatique. Nous savions que si nous prenions beaucoup de données, nous allions perdre du temps. Ce choix nous permettait donc d'être efficaces. Nous savions enfin que si l'encodage automatique fonctionnait sur ces deux séances, il serait alors possible de l'appliquer sur le restant du corpus, tout en restant vigilant sur la validité de chacun des fichiers XML-TEI créés.

6.3.2 Les étapes prérequises

Certaines étapes étaient à mettre en place avant de pouvoir construire les scripts et les appliquer. Au regard de la méthode de conversion choisie, nous devions commencer par annoter manuellement les données dans l'interface de l'outil du LRDE. Les deux coordinateurs du projet, Marie Puren et Pierre Vernus, se sont chargés de cette mission et ont inscrit les tags sur chaque zone segmentée de niveau paragraphe selon le guide d'annotation. En parallèle de cela, ils se sont assurés de la qualité de l'océrisation, et quand cela était nécessaire, ont resegmenté les zones puis réocérisé les textes. Par ailleurs, comme cela représentait déjà un gros travail, nous avons décidé de laisser de côté la reconnaissance des entités nommées, puisque celle-ci ne fonctionnait pas parfaitement bien et nécessitait une reprise supplémentaire pour pouvoir être exploitée. En plus de cet aspect pratique, l'équipe n'était pas encore fixée sur leur traitement et se demandait notamment s'il ne valait mieux pas tester d'autres outils que celui du LRDE pour les traiter le plus efficacement possible. L'objectif de cette mission était donc d'obtenir une sortie JSON avec les données textuelles correctes et annotées.

De plus, nous devions appliquer sur les sorties JSON un script python de remise en ordre des box, élaboré par le LRDE, car les box de ces fichiers n'étaient pas toujours dans le bon ordre. Cette erreur était due à un défaut de l'outil. Or, comme il était essentiel pour nous d'obtenir les box dans le bon ordre afin de respecter l'ordre de lecture du texte, les membres du LRDE ont mis en place ce script pour pallier le problème. À l'issue de cela, nous avions des fichiers JSON exploitables en fonction de nos besoins pour l'automatisation de l'encodage.

En outre, j'ai organisé l'environnement de travail pour la gestion automatique future des données. Pour cela, je me suis assurée que la convention de nommage des fichiers JSON était bien respectée. Le nommage des fichiers était en effet très important, car les scripts python allaient être configurés pour traiter les fichiers ayant pour nom le détail de la convention de nommage. J'ai créé également deux dossiers, « json_data » et « xml_data » [11], pour pouvoir stocker les fichiers JSON prêts à être traités et les fichiers futurs convertis en XML-TEI. Dans le dossier « xml_data », j'ai également intégré le

11. Se référer à l'annexe C.3.

schéma de l'encodage (fichier agoda_schema.rng) afin de pouvoir gérer dans un deuxième temps la validation des fichiers convertis. Enfin, j'ai choisi l'IDE [12] PyCharm [13] comme environnement pour développer l'ensemble des scripts. Il me permettait d'avoir une interface graphique fonctionnelle, avec notamment un éditeur de texte pour programmer, un bouton me permettant de démarrer le compilateur [14] et de visualiser le résultat de mon programme, l'accès direct au terminal, et un débogueur [15] pour gérer les erreurs plus facilement. J'ai aussi fait le choix de lier le dossier contenant les futurs scripts à un environnement virtuel. Cet environnement me permettait d'installer localement tous les paquets python nécessaires à mes scripts, sans qu'ils le soient sur l'ensemble de l'ordinateur.

12. *Integrated Development Environment* ou environnement de développement intégré.

13. *PyCharm*, JetBrains, URL : `https://www.jetbrains.com/fr-fr/pycharm/` (visité le 14/08/2022).

14. « Programme d'ordinateur qui traduit un programme en « langage machine ». ». Définition issue du site web suivant : `https://dictionnaire.lerobert.com/definition/compilateur`.

15. « Un débogueur, ou debugger, est un programme informatique permettant de détecter et de diagnostiquer les erreurs dans les logiciels. ». Définition issue du site web suivant : `https://www.ionos.fr/digitalguide/sites-internet/developpement-web/debogueur/`.

Chapitre 7

Le processus d'encodage automatique : analyse de la chaîne de traitement

Nous allons expliquer le fonctionnement concret du processus d'encodage automatique, en analysant les règles de transformation pour la conversion des données, et les différentes étapes de la chaîne de traitement. Les scripts mis en place sont construits à l'aide de la version 3.8.10 de Python et sont basés sur différentes librairies, et différents packages et modules, que nous citerons dans la suite de nos propos.

7.1 Du tag à la balise TEI : création des règles de transformation

Les règles de transformation ont été pensées pour convertir les données des fichiers JSON en données XML-TEI. Ces dernières s'appuient sur les tags associés aux textes, et sur le modèle d'encodage prédéfini. Nous allons évoquer d'abord l'organisation et la structure de ces dernières, puis nous étudierons plus précisément les différents types de règles élaborés.

7.1.1 Organisation et structure des règles

Une règle de transformation a pour rôle d'ajouter un élément TEI pour chaque box des fichiers JSON étant annotée à l'aide d'un tag particulier. Elle est construite en langage python, sous forme de fonction, c'est-à-dire sous forme de bloc de code regroupant plusieurs instructions et pouvant donner optionnellement une valeur de retour. Chaque fonction, autrement dit chaque règle de transformation, traite un ensemble de tags définis

ayant un même objectif d'encodage [1]. Par exemple, la fonction nommée « add_utterance » permet de traiter l'ensemble des étiquettes pensées pour annoter, avec la balise <u>, les prises de paroles (« u », « u-beginning » et « u-end »). Les fonctions sont construites de façon similaire et sont divisées en plusieurs instructions, que nous pouvons analyser au regard de l'exemple suivant.

```
def add_utterance(data):
    """
    Ajout de l'élément TEI "u" pour chaque boxe étiquetée "u" ou "u-beginning" et "u-end"
    :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"\bu(?!-)\b", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['<u>', data[i]['text_ocr'], '</u>'])
            elif re.search(r"u-beginning", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['<u>', data[i]['text_ocr']])
            elif re.search(r"u-end", data[i]["comment"]):
                data[i]['text_ocr'] = "".join([data[i]['text_ocr'], '</u>'])
            else:
                pass
    return data
```

FIGURE 7.1 – Script de la fonction add_utterance

La fonction add_utterance comprend une boucle contenant un ensemble de conditions. Une boucle est un procédé qui permet d'exécuter plusieurs fois un bloc de code tant qu'une condition est vérifiée. Dans notre cas, elle permet d'aller chercher une à une les boxes constitutives des fichiers JSON, de la première à la dernière. Ce procédé est ensuite conditionné, puisqu'il s'agit de ne retenir que les box contenant la clé « comment » avec une valeur. Lorsque cette condition est validée, chaque box contenant la clé « comment » est prise en compte et d'autres instructions conditionnelles sont appliquées. Ces dernières permettent de traiter plusieurs cas de figure. Il y a ici trois sous conditions permettant de donner trois instructions différentes :

— si la clé « comment » a pour valeur l'étiquette « u », alors il faut joindre au texte contenu dans la clé « text_ocr » la balise ouvrante <u> et la balise fermante </u>, à situer avant et après le texte ;
— ou si la clé « comment » a pour valeur l'étiquette « u-beginning », alors il faut joindre au texte contenu dans la clé « text_ocr » la balise ouvrante <u>, à situer avant le texte ;
— ou encore, si la clé « comment » a pour valeur l'étiquette « u-end », alors il faut joindre au texte contenu dans la clé « text_ocr » la balise fermante </u>, à situer après le texte.

1. Se référer au fichier guide_modelisation_encodage_automatique.pdf en annexe C.1.

Si la première condition n'est pas validée, alors le programme passe à la condition suivante et ainsi de suite. Lorsque l'ensemble des boxes ont été traitées, la fonction s'arrête et enregistre les résultats dans « data », c'est-à-dire la variable contenant l'ensemble des données balisées.

Afin de sélectionner les valeurs des clés « comment », j'ai utilisé le module « re » [2] de Python, celui-ci permettant d'utiliser les expressions régulières. Ces dernières sont un langage à part entière, intégré dans Python grâce à ce module. Elles correspondent à des motifs, autrement dit à des séquences de caractères, qui ont un sens particulier et qui permettent de rechercher des suites de caractères précises. Ce module était essentiel pour la construction des règles de transformation, car nous devions sélectionner des chaînes de caractères, les tags, puis les remplacer. Comme certains tags pouvaient avoir une construction similaire, il était essentiel de pouvoir réussir à sélectionner le bon. Dans l'exemple ci-dessus, le module « re » m'a permis, dans la première sous condition, de sélectionner l'étiquette « u » sans que les étiquettes « u-beginning » et « u-end », commençant toutes deux par le caractère « u », soient sélectionnées. Pour cela, j'ai utilisé les expressions régulières « \b » et « (?!-) » permettant de capturer le caractère « u » sans caractères ni devant ni derrière, et sans tiret à la suite de ce dernier. Cette recherche a été effectuée grâce à la méthode re.search(), celle-ci contenant comme paramètres le motif à sélectionner et l'endroit où chercher. Afin de gérer la sélection des autres étiquettes, j'ai dû adapter les expressions régulières en fonction des besoins. Je me suis aidée d'un éditeur regex [3] afin de les construire et de les tester.

Étant donné le nombre conséquent de tags, treize fonctions ont été élaborées afin de tous les prendre en compte. Afin de simplifier le fichier contenant l'ensemble de ces dernières, j'ai fait le choix de les répartir en trois fichiers distincts selon la fonction d'encodage des tags :

— le premier fichier regroupe les fonctions permettant l'ajout des balises formelles [4] ;
— le deuxième fichier contient celles permettant l'ajout des balises structurelles [5] ;
— le troisième fichier rassemble celles permettant l'ajout des balises sémantiques [6].

7.1.2 Trois types de règles

Comme nous venons de l'évoquer, les différentes fonctions ont pour rôle de transformer les données en ajoutant un élément TEI et sont toutes construites selon une structure similaire, utilisant une boucle initiale, des conditions, et le module « re ». Cependant, elles

2. *Re — Regular Expression Operations — Python 3.10.6 Documentation*, URL : `https://docs.python.org/3/library/re.html` (visité le 14/08/2022).
3. Dib (Firas), *Regex101 : Build, Test, and Debug Regex*, regex101, URL : `https://regex101.com/` (visité le 14/08/2022).
4. Se référer au fichier script_balisage_formel.py en annexe C.2.
5. Se référer au fichier script_balisage_structurel.py en annexe C.2.
6. Se référer au fichier script_balisage_semantique.py en annexe C.2.

se distinguent par les tags qu'elles traitent, mais aussi par la méthode de transformation effectuée. En effet, cette transformation pouvait correspondre à l'ajout des éléments TEI et pouvait être réalisée selon deux méthodes : soit par jointure (méthode Python .join()), soit par remplacement (méthode Python .replace()). La transformation pouvait, au contraire, correspondre à une suppression.

D'une part, la méthode d'ajout par jointure permettait d'intégrer une balise TEI avant et/ou après la valeur de la clé « text-ocr ». Cela correspond par exemple au cas de la fonction « add_utterance » évoqué précédemment. Nous devions insérer la balise <u> au début de la prise de parole, puis la balise fermante </u> à la fin de cette dernière. Pour cela, j'ai utilisé la méthode Python .join(). Elle me permettait de joindre les balises et le texte au sein d'une unique chaîne de caractères, celle-ci étant comprise comme la nouvelle valeur de la box. Cette méthode prend pour paramètres les valeurs à joindre.

```
{
 "activities": [],
 "addresses": [],
 "box": [
  57.6116701150507,
  1710.0,
  560.3883298849495,
  50.07766597698992
 ],
 "checked": true,
 "comment": "u seg",
 "id": 305,
 "key": [
  0,
  1735
 ],
 "ner_xml": "<PER>M. Paul Déroulède</PER>. Je demande la pa-<0x2029>Tole",
 "origin": "computer",
 "parent": 269,
 "persons": [
  "M. Paul Déroulède"
 ],
 "text_ocr": "M. Paul Déroulede. Je demande la pa-\nrole.",
 "type": "ENTRY"
},
```


(a) Exemple d'un extrait de fichier JSON avec une prise de parole

```
def add_utterance(data):
    """
    Ajout de l'élément TEI "u" pour chaque boxe étiquetée "u" ou "u-beginning" et "u-end"
    :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"\bu(?!-)\b", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['<u>', data[i]['text_ocr'], '</u>'])
            elif re.search(r"u-beginning", data[i]["comment"]):
                data[i]['text_ocr'] = "".join(['<u>', data[i]['text_ocr']])
            elif re.search(r"u-end", data[i]["comment"]):
                data[i]['text_ocr'] = "".join([data[i]['text_ocr'], '</u>'])
            else:
                pass
    return data
```

(b) Script de la fonction add_utterance

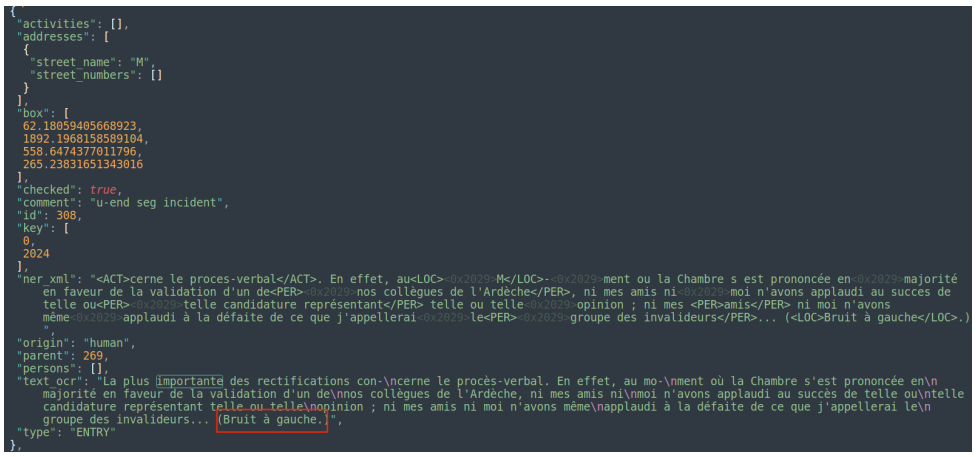
```
<u><seg>M. Paul Déroulede. Je demande la parole.</seg></u>
```

(c) Résultat de l'application de la fonction add_utterance

FIGURE 7.2 – Exemple d'application de la fonction add_utterance

D'autre part, la méthode d'ajout par remplacement permettait d'intégrer une balise TEI au sein même du texte. En effet, l'encodage spécifique des incidents et des citations nécessitait un traitement particulier. Ces deux cas pouvaient correspondre à un fragment du texte contenu dans la box. Je devais alors pouvoir insérer les balises ouvrantes et fermantes là où ils étaient situés précisément. Pour cela, je me suis aidée de chaînes de caractères comme repères afin d'insérer les balises. Comme chaque incident était encadré par des parenthèses, j'ai utilisé ce repère afin d'y ajouter la balise ouvrante et la balise fermante aux deux extrémités des parenthèses. La méthode .replace() m'a permis de traiter ce type de cas, prenant pour paramètres le caractère à remplacer (la parenthèse)

et les caractères de remplacements (les balises avec la parenthèse). Concernant le cas des citations, j'ai procédé de la même façon, mais en utilisant comme repère les guillemets.



(a) Exemple d'un extrait de fichier JSON avec un incident

```python
def add_incident(data):
    """
    Ajout des éléments TEI "incident" et "desc" au niveau des parenthèses ouvrantes et fermantes pour chaque boxe
    étiquetée "incident" ou "incident-beginning" et "incident-end"
    :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"\bincident(?!-)\b", data[i]["comment"]):
                data[i]['text_ocr'] = data[i]['text_ocr'].replace('(', '<incident><desc>(').replace(')',
                                                                                                   ')</desc></incident>')
            elif re.search(r"incident-beginning", data[i]["comment"]):
                data[i]['text_ocr'] = data[i]['text_ocr'].replace('(', '<incident><desc>(')
            elif re.search(r"incident-end", data[i]["comment"]):
                data[i]['text_ocr'] = data[i]['text_ocr'].replace(')', ')</desc></incident>')
            else:
                pass
    return data
```

(b) Script de la fonction incident

```
<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>
```

(c) Résultat de l'application de la fonction incident

FIGURE 7.3 – Exemple d'application de la fonction incident

Toutefois, cette méthode pouvait avoir des limites, puisque parfois le point de repère était omis. Or s'il manquait une parenthèse ou un guillemet ouvrant ou fermant, nous ne pouvions pas insérer de balises ouvrantes ou fermantes, ce qui aurait entraîné la non-validité du fichier XML. C'est pourquoi les cas où les points de repère étaient manquants n'ont pas été traités.

Enfin, à l'inverse des deux méthodes d'ajout, celle par suppression permettait de supprimer des données textuelles que je ne souhaitais pas inclure dans l'encodage. Nous avions créé pour cela le tag particulier « useless » permettant d'étiqueter ces informations inutiles (les pieds de page par exemple). Comme précédemment, j'ai mis en place une fonction permettant de gérer ces cas-ci et ai utilisé la méthode .join() afin de construire une chaîne vide, celle-ci étant la nouvelle valeur de la box.


```
{
 "activities": [],
 "box": [
  84.0,
  205.0,
  512.0,
  60.0
 ],
 "checked": true,
 "comment": "useless",
 "id": 606,
 "ner_xml": "",
 "origin": "computer",
 "persons": [],
 "text": "",
 "text_ocr": "CHAMBRE DES DEPUTES",
 "type": "TITLE_LEVEL_2",
 "key": [
  0,
  235
 ]
},
```


(a) Exemple d'un extrait de fichier JSON avec des données à supprimer

```
def delete(data):
    """
    Supprime les données ayant pour étiquette "useless"
    :param data:
    """
    for i in range(len(data)):
        if "comment" in data[i]:
            if re.search(r"useless", data[i]["comment"]):
                data[i]['text_ocr'] = "".join("")
            else:
                pass
    return data
```

(b) Script de la fonction delete

FIGURE 7.4 – Exemple de données inutiles à supprimer avec le script delete

7.2 Des fichiers JSON aux fichiers XML-TEI : les étapes de la chaîne de traitement

Afin de passer des fichiers JSON aux fichiers XML, j'ai développé une chaîne de traitement d'automatisation qui comprend trois étapes principales. Ces trois phases sont réunies au sein d'un script général appelé « main » [7] qui permet de les appliquer successivement.

7.2.1 Création des fichiers composants

La première phase de la chaîne de traitement consiste en la création des fichiers composants. Elle se divise en six étapes :

— Étape 1 : récupération des données JSON. Celles-ci sont stockées dans une variable « data » sous forme d'un dictionnaire. Par la suite, c'est sur cette variable que nous interagissons afin d'appliquer un ensemble d'instructions ;

— Étape 2 : gestion de l'encodage des changements de page. Comme il s'agit d'un traitement particulier, celui-ci se fait en amont. En effet, je crée une variable qui prend en valeur initiale le numéro de page de la première page du compte rendu et qui s'incrémente automatiquement à chaque changement de page ;

— Étape 3 : intégration des scripts contenant les règles de transformation [8] dans la chaîne de traitement. Ils permettent d'encoder les données contenues dans la variable « data ». Ces fonctions sont appelées dans un ordre bien précis au sein de la fonction « compilation » qui permet de garder en mémoire les résultats des applications des scripts sur data. L'ordre des fonctions préétabli permet d'appliquer les balises convenablement suivant l'ordre hiérarchique de l'arbre XML, tout en évitant des chevauchements de balises ;

— Étape 4 : gestion des métadonnées [9]. Afin d'automatiser la construction de l'élément <teiHeader> contenant les métadonnées, cette étape se divise en deux temps :

- d'une part, la rédaction d'une partie du <teiHeader> à la main contenant les métadonnées fixes, c'est-à-dire qui ne varient pas entre les différents comptes rendus. Concernant les métadonnées variables, celles-ci seront appelées lors de la deuxième sous-étape ;
- d'autre part, la construction de règles permettant de rechercher les métadonnées propres à chaque compte rendu, présentes dans la variable « data », qui sont par la suite intégrée au <teiHeader> préalablement établi.

7. Se référer au fichier main.py en annexe C.2.
8. Se référer au fichier script_compilation.py en annexe C.2.
9. Se référer au fichier script_metadonnees.py en annexe C.2.

— Étape 5 : création des fichiers XML. Toutes les règles ayant été appliquées à la variable « data », le résultat de celle-ci est enregistré dans un fichier XML. De plus, on y compile l'élément racine TEI qui est enregistré dans une variable, le <teiHeader> contenant l'ensemble des métadonnées, et enfin la balise fermante </TEI> permettant de clore le fichier XML ;
— Étape 6 : nettoyage des fichiers XML. Cette dernière étape permet de recoller les mots qui ont été divisés à cause des caractères \n, ces derniers signifiant un saut de ligne en Python [10].

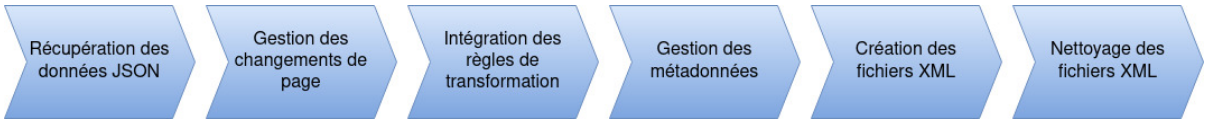


FIGURE 7.5 – Schéma de la chaîne de création des fichiers composants

7.2.2 Création du fichier corpus

La deuxième phase de la chaîne de traitement porte sur la création du fichier corpus. Celui-ci contient les métadonnées propres au corpus contenant l'ensemble des comptes rendus d'une législature. Il permet, de plus, de rassembler l'ensemble des fichiers XML composants. Sa création se fait en deux étapes :

— Étape 1 : génération du fichier corpus. Comme le fichier corpus de la législature traitée avait déjà été construit lors de l'encodage test, je n'ai pas inclus la génération du fichier dans la chaîne de traitement d'automatisation. En effet, la réflexion et le développement des scripts portant sur une seule législature, l'automatisation de cette étape n'était pas utile à ce stade du projet ;
— Étape 2 : intégration des métadonnées. Ces dernières, tout comme la génération du fichier corpus, ont été rédigées à la main, lors de l'encodage test. Je n'ai donc pas eu besoin d'intégrer cette étape dans la chaîne de traitement d'automatisation ;
— Étape 3 : gestion des <xi :include>. Afin de compiler l'ensemble des fichiers XML composants au sein du fichier corpus, j'ai utilisé la librairie Python *lxml* [11], spécifique pour la gestion du format XML, dans le but d'intégrer, à la suite des métadonnées, les marqueurs <xi :include>. J'ai créé une boucle qui peut permettre d'aller chercher les noms des fichiers XML composants, afin de les associer au fichier XML corpus à l'aide à l'aide de l'élément <xi :include>.

10. Se référer au fichier script_nettoyage.py en annexe C.2.
11. *Lxml - Processing XML and HTML with Python*, URL : `https://lxml.de/` (visité le 14/08/2022).

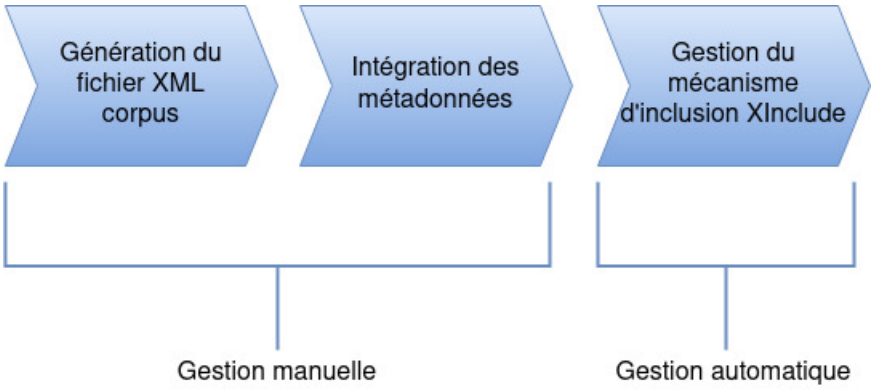


FIGURE 7.6 – Schéma de la chaîne de création du fichier corpus

7.2.3 Validation des fichiers XML-TEI

La troisième phase de la chaîne de traitement permet de vérifier la validation des fichiers XML construits par rapport au schéma d'encodage établi. Celui-ci a été intégré dans les déclarations du fichier XML corpus au moment de sa génération. Par la suite, je n'avais plus qu'à ouvrir le fichier dans l'éditeur XML Oxygen afin de vérifier la validité. Cependant, cette vérification n'a pas encore été incluse dans la chaîne de traitement d'automatisation.

Chapitre 8

Un processus concluant ?

Suite à la mise en place des règles de transformation et de la chaîne de traitement d'automatisation, j'ai souhaité évaluer le fonctionnement de ces dernières en appliquant le processus sur les deux comptes rendus sélectionnés en amont. Cette phase de tests m'a permis d'ajuster les scripts en fonction des erreurs, et de déterminer des points futurs d'amélioration.

8.1 Analyse de l'encodage obtenu

8.1.1 Résultats

Après divers ajustements, le processus d'automatisation a donné un résultat concluant pour l'encodage du compte rendu de la séance du 26 novembre 1889 [1]. J'ai pu obtenir un fichier XML composant contenant les métadonnées du document et l'ensemble du contenu textuel encodé selon une partie du modèle d'encodage prédéfini. Le fichier obtenu était conforme au format XML et valide selon le schéma d'encodage. Celui-ci était également relié au fichier XML corpus, grâce au mécanisme d'inclusion <xi :include>. Ce résultat m'a permis de constater que toutes les étapes de la chaîne de traitement fonctionnaient. Elles pourront donc être appliquées sur les autres comptes rendus du corpus, similaires à celui-ci.

Concernant le compte rendu de la séance du 14 janvier 1890, le résultat a été moins concluant. Le fichier XML obtenu n'était pas conforme au XML, ni valide selon le schéma d'encodage puisque les métadonnées et l'encodage du contenu textuel étaient lacunaires. La cause de ces manques était due à plusieurs raisons. La structure du compte rendu différait des autres, car il contenait plusieurs séances, et l'aspect formel de la première page différait aussi. Ces deux points devaient donc avoir un traitement particulier, qui m'a contraint d'élaborer sur le tard des étiquettes supplémentaires pour pouvoir leur appliquer un encodage spécifique. Ces dernières nécessitaient par la même de modéliser de nouvelles

1. Se référer au dossier /xml_data en annexe C.3.

règles de transformation. Au vu du temps qu'il me restait, j'ai préféré mettre de côté la gestion de ces particularités, d'autant plus que l'annotation effectuée dans les fichiers JSON était fautive à certains endroits. Le résultat de l'automatisation de ce compte rendu n'a pas été sauvegardé.

8.1.2 Un encodage à compléter

L'ensemble des règles de transformation que j'ai mises en place traitent une grande partie du modèle d'encodage prédéfini, mais elles ne prennent pas en compte l'intégralité de ce dernier. Elles doivent donc être complétées afin de pouvoir obtenir les comptes rendus encodés selon le modèle d'encodage idéal.

```
<!-- PARTIE 1-->
<div type="part" corresp="#pv">

   <head>Présidence de <persName ref="#pers_ID">M. Charles Floquet</persName></head>

   <note type="opening" xml:id="CR_1889-11-26_n1"><seg xml:id="CR_1889-11-26_n1.1">La
         séance est ouverte à <time when="02:00:00">deux heures</time>.</seg></note>

   <note type="comment" xml:id="CR_1889-11-26_n2"><seg xml:id="CR_1889-11-26_n2.1"
           ><persName ref="#pers_ID">M. Henri Lavertujon, l'un des <roleName
             ref="#pers_ID">secrétaires</roleName></persName>, donne lecture du
         procès-verbal de la séance d'hier.</seg></note>


   <u who="#pers_ID" xml:id="CR_1889-11-26_u1" ana="#speaker">
      <seg xml:id="CR_1889-11-26_u1.1"><persName ref="#pers_ID">M. Paul
            Déroulède</persName>. Je demande la parole.</seg>
   </u>

   <u who="#pers_ID" xml:id="CR_1889-11-26_u2" ana="#chair">
      <seg xml:id="CR_1889-11-26_u2.1"><persName ref="#pers_ID">M. le <roleName
               ref="#pers_ID">président</roleName></persName>. Vous avez la parole.</seg>
   </u>

   <u who="#pers_ID" xml:id="CR_1889-11-26_u3" ana="#speaker">
      <seg xml:id="CR_1889-11-26_u3.1"><persName ref="#pers_ID">M. Paul
            Déroulède</persName>. Messieurs, j'ai demandé la parole pour faire une
         rectification au procès-verbal et pour m'expliquer au sujet du rappel à l'ordre
         qui m'a été infligé.</seg>
      <seg xml:id="CR_1889-11-26_u3.2">La plus importante des rectifications concerne le
         procès-verbal. En effet, au moment où la <orgName ref="#org_ID">Chambre</orgName>
         s'est prononcée en majorité en faveur de la validation d'un de nos collègues de
            l'<placeName ref="#lieu_ID">Ardèche</placeName>, 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>
```

(a) Encodage idéal

```
<!-- PARTIE 1 -->
<div type="part">
    <head>PRESIDENCE DE M. CHARLES FLOQUET.</head>
    <note type="opening"><seg>La séance est ouverte à deux heures.</seg></note>
    <u><seg>M. Henri Lavertujon, l'un des secrétaires, donne lecture du
            procès-verbal de la séance d'hier.</seg></u>
    <u><seg>M. Paul Déroulede. Je demande la parole.</seg></u>
    <u><seg>M. le président. Vous avez la parole.</seg></u>
    <u><seg>M. Paul Déroulède. Messieurs, j'ai demandé la parole pour faire une
            rectification au procès-verbal et pour m'expliquer au sujet du rappel à
            l'ordre qui m'a été infligé.</seg>
        <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>
```

(b) Résultat de l'encodage automatique

FIGURE 8.1 – Comparaison entre le modèle d'encodage idéal et le résultat obtenu

Tout d'abord, il faudra ajouter dans les règles de transformation la gestion des entités nommées lorsque la méthode de reconnaissance de celles-ci sera adoptée. En fonction de cette dernière et du résultat obtenu, il sera possible de modéliser de nouvelles règles de transformation, ou de les intégrer directement dans la chaîne de traitement. De plus, tous les attributs précisés dans le modèle d'encodage n'ont pas été traités : les attributs @xml :id, @who, @ana, et @correp devront être ajoutés au sein de certains éléments. Ces derniers ont été mis de côté, car soit il nécessitait une gestion particulière, soit il n'était pas encore possible de les traiter. Par exemple, les @xml :id, ayant pour valeur une numérotation, devront faire l'objet d'une incrémentation automatique, afin de numéroter chaque prise de parole et paragraphe. L'attribut @who, quant à lui, devra prendre pour valeur l'identifiant des orateurs. Il faudra alors définir l'ensemble des identifiants, et trouver un moyen pour les intégrer au bon endroit automatiquement. L'inclusion des attributs au sein de l'encodage déjà généré pourra faire l'objet d'une deuxième phase d'encodage, traitée à part des règles de transformation. Elle pourra, en effet, être effectuée à l'aide de la librairie *lxml* spécifique à la gestion du format XML. Cette dernière permettra de parser le fichier XML et d'y inclure à l'endroit souhaité les attributs.

Par ailleurs, au vu de la complexité de certains éléments d'encodage, j'ai décidé d'appliquer un modèle simplifié lors de la modélisation des règles de transformation. Cela concerne spécifiquement l'encodage des tableaux.

```
<table rows="4" cols="2" corresp="vot18891126_vot1">
   <row>
      <cell role="label">Nombre des votants</cell>
      <cell role="data"><num>514</num></cell>
   </row>
   <row>
      <cell role="label">Majorité absolue</cell>
      <cell role="data"><num>258</num></cell>
   </row>
   <row>
      <cell role="label">Pour l'adoption</cell>
      <cell role="data"><num>333</num></cell>
   </row>
   <row>
      <cell role="label">Contre</cell>
      <cell role="data"><num>181</num></cell>
   </row>
</table>
```

(a) Encodage idéal tableau 1

```
<table>
    <row>
        <cell>Nombre des votants..... 514 Majorité absolue.....
            258 Pour l'adoption..... 333 Contre..... 181 </cell>
    </row>
</table>
```

(b) Résultat tableau 1

FIGURE 8.2 – Comparaison entre l'encodage idéal et le résultat obtenu pour le premier modèle de tableau

```
<desc>
   <measure type="nbvoters" quantity="506">Nombre des votants
      <num>506</num></measure>
   <measure type="majority" quantity="254">Majorité absolue <num>254</num></measure>
   <measure type="ayes" quantity="330">Pour l'adoption <num>330</num></measure>
   <measure type="noes" quantity="176">Contre <num>176</num></measure>
</desc>
```

(a) Encodage idéal tableau 2

```
<table>
    <row>
        <cell>Nombre des votants..... 506 Majorité
            absolue..... 254 Pour l'adoption..... 330
            Contre..... 176 </cell>
    </row>
</table>
```

(b) Résultat tableau 2

FIGURE 8.3 – Comparaison entre l'encodage idéal et le résultat obtenu pour le second modèle de tableau

Deux modèles d'encodage étaient préconisés initialement pour leur balisage. Il y avait le cas de la gestion des tableaux dans le corps du texte, comprenant l'élément <table> contenant un ensemble de colonnes (<row>), elles-mêmes composées de cellules (<cell>), et le cas de la gestion des tableaux au sein des annexes, comprenant l'élément <desc> contenant plusieurs <measure>. Je n'ai pas pris en compte la distinction de ces deux modèles et j'ai appliqué pour tous les cas le premier modèle. En plus de cette simplification, et comme l'ensemble du texte était contenu dans une seule box du fichier JSON, je n'ai pas respecté l'encodage en tout point et j'ai inclus l'ensemble des données au sein d'une seule cellule. Il s'agira donc d'une part d'améliorer ce point en répartissant le texte correctement en fonction des colonnes et des cellules, et d'autre part, il faudra retraiter les tableaux des annexes, pour leur appliquer le deuxième modèle d'encodage.

Enfin, il faudra traiter les spécificités de la première page du compte rendu du 14 janvier 1890, mis de côté jusqu'alors. Comme ce cas particulier ne concerne qu'un très petit nombre de comptes rendus, l'inclusion de leur gestion dans le processus d'encodage automatique pourra être questionnée. Il sera intéressant, en effet, d'évaluer les avantages de l'automatisation par rapport aux besoins techniques qu'ils nécessitent.

8.2 Analyse des possibilités d'amélioration de la chaîne de traitement

Le résultat de la chaîne de traitement d'automatisation est concluant puisque comme nous venons de le voir, nous obtenons des fichiers composants XML conformes, valides, et reliés au fichier corpus. Suite à son application et à l'évaluation de son résultat, nous

avons pu toutefois constater quelques manques et inconvénients. Nous allons voir quelles sont les clés d'amélioration possibles de celle-ci.

8.2.1 Automatiser l'annotation des fichiers JSON

D'une part, l'annotation manuelle au sein de l'interface de l'outil d'OCR du LRDE a été une étape très longue et difficile à effectuer. La complexité du guide d'annotation et la longueur des comptes rendus rendaient la tâche très compliquée. En effet, de nombreuses relectures et corrections ont dû être réalisées après la première annotation, car il y a avait beaucoup d'erreurs. Or, comme cette étape était primordiale au processus d'automatisation puisque l'ensemble des règles de transformation reposait sur cette dernière, il fallait absolument que l'annotation soit de qualité parfaite afin de pouvoir obtenir des fichiers XML conformes et valides. En plus de ce constat, il n'était pas pensable d'appliquer les étiquettes à la main sur l'ensemble des comptes rendus du corpus.

L'objectif serait donc d'inclure cette étape dans la chaîne d'automatisation afin de gagner en efficacité et d'assurer la qualité du résultat. Une première solution a été pensée pour être appliquée à court terme. Elle consistera à construire un script Python permettant de corriger automatiquement l'orthographe des étiquettes. En effet, la majorité des corrections effectuées sur l'annotation reposaient sur des erreurs orthographiques de ces dernières. Le script pourrait donc être un bon moyen de limiter ces erreurs d'orthographe. Cependant, il ne permettra pas d'annoter automatiquement les comptes rendus. C'est pourquoi il faudra mettre en place une deuxième solution, applicable sur le long terme, permettant d'automatiser l'annotation. Cette automatisation garantira, par la même, la qualité du résultat. Elle pourra être mise en place grâce à l'intelligence artificielle, et notamment avec le *machine learning*, autrement dit l'apprentissage automatique. En effet, l'objectif sera de faire apprendre à l'ordinateur la structure et les éléments sémantiques des comptes rendus, pour qu'il puisse placer les étiquettes automatiquement.

8.2.2 Automatiser la gestion des erreurs

D'autre part, lors des premières applications du processus d'automatisation, j'ai été confrontée à de nombreuses erreurs et ai mis du temps pour toutes les localiser. Elles pouvaient être dues soit à une erreur d'annotation impactant le résultat des conversions, soit au code lui-même. Je me suis retrouvée alors à devoir trouver l'origine de ces nombreuses erreurs pour pouvoir les corriger. Afin de gagner en efficacité, il serait utile d'intégrer au sein des scripts, des tests unitaires. Ces derniers pourraient alors vérifier le bon fonctionnement du programme et préciser à quel niveau du code l'erreur intervient. Il serait aussi utile d'inclure, comme prévu initialement, la validation automatique des fichiers XML-TEI par rapport au schéma Relax NG. Cette dernière doit être effectuée pour l'instant à la main, en consultant directement les fichiers dans l'éditeur XML. Or cela est contrai-

gnant, car lorsque le processus d'automatisation sera appliqué sur l'ensemble du corpus, il faudra consulter chaque fichier un à un pour localiser les erreurs. L'objectif serait dons de pouvoir automatiser cette tâche grâce à la librairie *lxml*, afin que la localisation des erreurs soit indiquée directement dans l'interface de programmation.

Conclusion

Ce présent mémoire s'est attaché à décrire le travail réalisé au cours du stage que j'ai effectué au sein de l'équipe du projet AGODA. La mission qui m'a été confiée consistait en la structuration et l'enrichissement sémantique des comptes rendus *in extenso* des débats parlementaires de la Chambre des députés de la V^{e} législature de la III^{e} République (1889-1893). Les réflexions que j'ai menées pour réaliser à bien cette mission se sont centrées sur les caractéristiques de la source à traiter, sur les technologies à utiliser, et sur les méthodes de traitement à appliquer, au regard des objectifs scientifiques et techniques de la chaîne de traitement du projet. La réalisation de cette mission s'est déroulée en plusieurs étapes successives illustrées au sein des trois parties de ce mémoire.

La première partie a consisté à appréhender la source des comptes rendus *in extenso* des débats parlementaires, en étudiant leur contexte de production, en définissant les caractéristiques propres à cette typologie documentaire, et en analysant les différents modes de consultation de cette source et les problématiques qui leur sont liés. Cette étape préliminaire était essentielle pour saisir le cœur du projet AGODA. Celui-ci s'inscrit, en effet, comme réponse aux difficultés rencontrées par les chercheurs et le grand public pour explorer les débats parlementaires et s'attache pour cela à faciliter l'accessibilité et l'exploitabilité de ces derniers. La présentation des objectifs d'AGODA a permis, ensuite, de déterminer le cadre dans lequel s'inscrivait ma mission, celle-ci ayant une place centrale dans la chaîne de traitement. Située en aval de l'océrisation des comptes rendus, et en amont de la publication de ces derniers, cette étape d'annotation et d'enrichissement sémantique était capitale pour répondre aux enjeux du projet. Elle nécessitait aussi de prendre en considération, pour sa réalisation, les spécificités de l'étape d'océrisation.

La deuxième partie a permis d'étudier l'élaboration du modèle d'encodage XML-TEI, spécifiquement pensé pour la typologie des comptes rendus des débats parlementaires. Conçus dans le but d'obtenir des données structurées et enrichies, les principes d'encodage ont été constitués selon les objectifs scientifiques du projet, et en fonction d'un cadre technique et pratique bien défini. Ils reposent sur les aspects physique, logique et sémantique des débats, et ont été formalisés par un schéma au format Relax NG et documentés au sein d'un ODD. Ce processus de modélisation a accordé une grande importance à la standardisation des données, grâce au choix de l'XML-TEI, au respect de cette norme, mais aussi grâce à la prise en compte des modèles d'encodage existants sur cette

même typologie documentaire. Il a répondu, par la même, aux principes de réutilisabilité et d'échangeabilité des données.

Enfin, la troisième partie avait pour objectif de présenter le processus d'encodage automatique mis en place pour baliser l'ensemble du corpus selon le modèle d'encodage prédéfini. L'analyse des outils à disposition nous a permis de justifier les différents choix méthodologiques effectués. Nous avons pu également étudier les différents scripts Python, permettant la conversion des données, et leur gestion en tant que corpus, ainsi que le résultat obtenu suite à l'application du processus. Ce dernier nous a, en effet, permis d'obtenir un compte rendu encodé automatiquement, conforme et valide au modèle d'encodage défini. Nous avons également mentionné les limites de l'automatisation, en démontrant que tout n'était pas automatisable, et que nous avions privilégié sur certains points un balisage à la main.

La réalisation de ces différentes tâches m'aura permis de découvrir les comptes rendus *in extenso* des débats parlementaires, d'approfondir mes connaissances sur différentes technologies numériques, tout en me questionnant sur les apports de ces dernières pour répondre aux enjeux du projet. Ce stage aura été aussi l'occasion pour moi de présenter mon travail lors de plusieurs communications scientifiques.

En conclusion, de nombreuses perspectives se dessinent suite à l'issue de mon stage. Bien que certains points devront être complétés et améliorés, le processus de balisage automatique pourra être appliqué sur l'ensemble du corpus afin d'obtenir des données structurées et enrichies pour l'ensemble des comptes rendus de la V^{e} législature de la III^{e} République. Celles-ci pourront être publiées ensuite sur la plateforme *TEI Publisher*. Cette étape d'éditorialisation devra être définie concrètement, à partir du modèle d'encodage élaboré, afin de mettre en place les fonctionnalités de visualisation et d'exploitation du corpus sur la plateforme de consultation. L'équipe du projet AGODA pourra, *in fine*, élargir son champ d'action en traitant le corpus prévu initialement, correspondant à l'ensemble des comptes rendus de la Chambre des députés de la III^{e} République (1881-1940). Ainsi, le projet AGODA est sur la voie pour réaliser son objectif premier : l'ouverture et l'exploration des débats parlementaires.

Annexes

Les annexes accompagnant le présent mémoire correspondent aux livrables techniques et sont consultables sur un *repository* Github, situé à l'adresse suivante : `https://github.com/FannyLbr/Memoire-AGODA-TNAH2022.git`

Cette section contient les indications sur la localisation des fichiers dans les dossiers d'annexes.

Annexe A

Sources primaires

A.1 Numérisations Gallica

Le dossier /A - /A1 - Numérisations Gallica/ contient les numérisations de deux comptes rendus des débats parlementaires, celui de la séance du 26 novembre 1889 et celui de la séance du 14 janvier 1890 :

— FR_3L_5L_1889-11-26_digitisation.pdf

— FR_3L_5L_1890-01-14_digitisation.pdf

A.2 Images Archives nationales

Le dossier /A - /A2 - Images Archives nationales/ contient des photographies prises lors d'une visite aux Archives nationales du compte rendu *in extenso* et du procès-verbal de la séance du 26 novembre 1889 :

— Dans le dossier /CR, voir :

- 20220505_110636.jpg
- 20220505_111410.jpg
- 20220505_111418.jpg
- 20220505_111438.jpg
- 20220505_111448.jpg
- 20220505_111450.jpg

— Dans le dossier /PV, voir :

- 20220505_102611.jpg
- 20220505_102621.jpg
- 20220505_102631.jpg
- 20220505_102641.jpg
- 20220505_102648.jpg
- 20220505_102657.jpg

- 20220505_102748.jpg
- 20220505_102752.jpg
- 20220505_102858.jpg

Annexe B

Modélisation XML-TEI

B.1 Documents de travail

Le dossier /B - /B1 - Documents de travail/ contient trois anciennes versions de l'encodage XML-TEI effectuées lors de la modélisation du modèle, ainsi qu'un document en *markdown* relatant les étapes de la construction du schéma :

— FR_3R_5L_1889-11-26_v1.xml
— FR_3R_5L_1889-11-26_v2.xml
— FR_3R_5L_1889-11-26_v3.xml
— schema_working_document.md

B.2 Encodage test

Le dossier /B - /B2 - Encodage test/ contient les modèles d'encodage idéaux du fichier XML-TEI corpus et du fichier XML-TEI composant, ainsi que le schéma d'encodage au format Relax NG qui est relié aux deux fichiers XML-TEI :

— FR_3R_5L.xml
— FR_3R_5L_1889-11-26_ideal_encoding_model.xml
— agoda_schema.rng

B.3 Schéma et documentation

Le dossier /B - /B3 - Schéma et documentation/ contient l'ODD aux formats XML et HTML, le schéma d'encodage au format Relax NG et les spécifications du modèle au format XML :

— agoda_odd.html
— agoda_odd.xml
— agoda_schema.rng
— agoda_schemaSpec.xml

Annexe C

Encodage automatique

C.1 Guides

Le dossier /C - /C1 - Guides/ contient le guide d'annotation et celui de l'encodage automatique :

— guide_annotations_agoda.pdf
— guide_modelisation_encodage_automatique.pdf

C.2 Scripts Python

Le dossier /C - /C2 - Scripts Python/ contient l'ensemble des scripts Python utilisés pour l'encodage automatique des données :

— main.py
— script_balisage_formel.py
— script_balisage_logique.py
— script_balisage_semantique.py
— script_compilation.py
— script_metadonnees.py
— script_nettoyage.py

C.3 Données tests

Le dossier /C - /C3 - Données tests/ contient l'ensemble des fichiers JSON annotés de la séance du 26 novembre 1889, ainsi que le résultat XML-TEI obtenu suite à l'application du processus d'encodage automatique :

— Dans le dossier /json_data, voir :
  - FR_3R_5L_1889-11-26_p0175.json
  - FR_3R_5L_1889-11-26_p0176.json

- FR_3R_5L_1889-11-26_p0177.json
- FR_3R_5L_1889-11-26_p0178.json
- FR_3R_5L_1889-11-26_p0179.json
- FR_3R_5L_1889-11-26_p0180.json
- FR_3R_5L_1889-11-26_p0181.json
- FR_3R_5L_1889-11-26_p0182.json
- FR_3R_5L_1889-11-26_p0183.json
- FR_3R_5L_1889-11-26_p0184.json
- FR_3R_5L_1889-11-26_p0185.json
- FR_3R_5L_1889-11-26_p0186.json
- FR_3R_5L_1889-11-26_p0187.json
- FR_3R_5L_1889-11-26_p0188.json
- FR_3R_5L_1889-11-26_p0189.json
- FR_3R_5L_1889-11-26_p0190.json
- FR_3R_5L_1889-11-26_p0191.json
- FR_3R_5L_1889-11-26_p0192.json
- FR_3R_5L_1889-11-26_p0193.json

— Dans le dossier /xml_data, voir :

- FR_3R_5L.xml
- FR_3R_5L_1889-11-26.xml
- agoda_schema.rng

Liste des acronymes

AGODA Analyse sémantique et Graphes relationnels pour l'Ouverture et l'étude des Débats à l'Assemblée nationale. xiv, xv, 17–25, 27, 28, 30–33, 37, 39, 40, 42, 43, 45, 50, 59, 60, 71, 83

ALMAnaCH Automatic Language Modelling and Analysis & Computational Humanities. 21

ANR Agence nationale de la recherche. 23, 30

API Application Programming Interface. 24, 28

BnF Bibliothèque nationale de France. 19, 20, 28, 32

CNRS Centre national de la recherche scientifique. 20

CSS Cascading Style Sheets. 79

DTD Document Type Definition. 71

Epita Ecole d'ingénieurs informatique Paris. 21

Epitech European Institute of Technology. xv, 20

EPUB Electronic Publication. 72, 73

FAIR Findable, Accessible, Interoperable, Reusable. 19, 24, 71

HTML HyperText Markup Language. 24, 72, 73, 78, 79, 132

HTR Handwritten text recognition. 28

HumaNum Infrastructure de recherche dédiée aux lettres, sciences humaines et sociales et aux humanités numériques. 19, 20, 29

IDE Integrated Development Environment. 99

Inria Institut national de recherche en informatique et en automatique. 21

JPEG Joint Photographic Experts Group. 15

JSON JavaScript Object Notation. 31, 86–90, 92–96, 98, 101, 102, 105–108, 112, 114, 132

OCR Optical Character Recognition. 27–31, 86, 89, 95, 115, 131

ODD One Document Does it All. 71–76, 78, 79, 132

ODT OpenDocument Text. 72, 73

PDF Portable Document Format. 15, 72, 73

Relax NG Regular Language for XML Next Generation. 71, 72, 76, 77, 115

sed Stream EDitor. 86

SGML Standard Generalized Markup Language. 40, 41

SoDUCo Social Dynamics in Urban Context. 30

TAL Traitement Automatique des Langues. 21, 32

TEI Text Encoding Initiative. xiv, 21, 23, 24, 33, 41–45, 49–53, 55, 56, 65, 71–76, 78, 87, 92, 93, 95, 101, 103–105, 109, 131

TXT Text File. 16

URI Uniform Resource Identifier. 19

W3C World Wide Web Consortium. 40

XML eXtensible Markup Language. 24, 27, 40–42, 44, 49–51, 54, 59, 60, 67, 68, 71, 72, 74, 76, 78, 85, 87, 88, 92–94, 106, 108–111, 113–115, 132

Table des figures

Table des matières