Modélisation de l'encodage automatique des débats parlementaires : transformation JSON vers XML TEI

1- Fonctionnement général de l'encodage automatique

Il est nécessaire au préalable de :

- Créer un dossier json_data pour stocker l'ensemble des fichiers JSON annotés avec les étiquettes, corrigés et dont les boxes sont dans le bon ordre.
- Créer un dossier xml_data pour stocker le schéma de l'encodage (agoda_schema.rng) et le fichier corpus (ex : FR_3R_5L.xml) rédigé à la main sans les inclusions (xi:include). Les fichiers composants XML balisés en TEI seront inclus dans ce dossier et les inclusions seront ajoutées automatiquement dans le fichier corpus.

Une fois cela effectué, il est possible de faire tourner les scripts. Les fichiers composants XML sont créés, les données sont balisées en XML TEI. Les inclusions des fichiers composants dans le fichier XML corpus sont effectuées.

Pour plus de lisibilité des fichiers XML, il est possible de les ouvrir manuellement et de les indenter à l'aide d'un éditeur XML.

Les données obtenues sont encodées en UTF-8.

2- Note sur cette présente modélisation

La modélisation est pensée pour :

- l'ensemble des CR de la 5e législature de la IIIe république (1889-1893).
- excepté les cas particuliers de la première séance de chaque année (CR à traiter par la suite selon ses spécificités et à inclure dans cette présente modélisation).

Certaines modifications et améliorations seront à effectuer sur les scripts lorsqu'il s'agira de traiter d'autres corpus de débats (évolutions des étiquettes).

Cette modélisation utilise des étiquettes pour appliquer les balises TEI de façon automatique. Ces dernières sont pensées en amont et incluses manuellement (pour l'instant) dans les fichiers JSON, au niveau de la clef "comment". Toutes les boxes contenant la clef "text_ocr" avec les données textuelles doivent être annotées à l'aide de ces étiquettes.

3- Tableaux descriptifs de la modélisation des scripts

Les tableaux présentés ci-dessous détaillent, script par script, les méthodes utilisées pour traiter l'ensemble des étiquettes que nous utilisons pour baliser le texte :

- chaque étiquette est associée à un résultat XML TEI attendu,
- certaines étiquettes peuvent avoir un traitement particulier qui est notifié,
- certaines étiquettes nécessitent d'être complétées (prise en compte des attributs XML par exemple).

❖ *Script balisage formel*

	Étiquettes	Résultat XML TEI	Méthode et incertitudes	À ajouter
add_seg **(paragraphes)**	seg	<seg>text</seg>	**Attention :** traitement particulier pour les seg couplés avec quote-beginning et quote-end car chevauchement de balises. Pour ces cas, le seg est traité dans la	**Ajouter :** @xml:id

			fonction add_quote.	
	seg-beginning	<seg>text		
	seg-end	text</seg>		
add_signed **(signature)**	signed	<signed> <seg>text</seg> </signed>		
add_page_number **(changements de pages)**	body	<pb n=””/>	Première page de la séance : à récupérer dans le texte. La récupération de la page permet de supprimer le restant de la ligne qui ne nous sert pas.	
	body1	<pb n=”1”/>	**Attention :** traitement différent à opérer pour la première page de l’année, le numéro de page est présent en bas de page.	**À inclure plus tard car traitement spécifique**
	page-number	<pb n=”(zwt +1)”/>	Numéro de la page à obtenir grâce à l’incrémentation effectuée à partir de la première page.	
	page-number-ref	<ref target="#(zwt +1)"/>		

			La récupération de la page permet de supprimer le restant de la ligne qui ne nous sert pas.	

❖ *Script balisage sémantique*

	Étiquettes	Résultat XML TEI	Méthode et incertitudes	À ajouter
add_utterance **(énoncés)**	u	<u><seg>Texte</seg></u>		**Ajouter :** @xml:id **Ajouter plus tard :** @who @ana
	u-beginning	<u><seg>Texte</seg>		
	u-end	<seg>Texte</seg></u>		
add_comment **(commentaires généraux)**	comment	<note type="comment"> <seg>Texte</seg> </note>		**Ajouter :** @xml:id **Ajouter plus tard :** @corresp pour les notes des rectifications
	comment-beginning	<note type=”comment”> <seg>Texte</seg>		
	comment-end	<seg>Texte</seg> </note>		
	result	<note type=”result”>		**Ajouter :** @xml:id

		<seg>Texte</seg> </note>		
	opening	<note type="opening"> <seg>Texte</seg> </note>		
	closing	<note type="closing"> <seg>Texte</seg> </note>		
add_incident **(commentaires atmosphères)**	incident	<incident><desc>(*Texte*)</desc></incident>	Utilisation d'un point de repère pour placer les balises sur le texte en question →parenthèses **Attention :** certains incident ne sont pas annotés car le point de repère est manquant dans l'OCR.	
	incident-beginning	<incident><desc>		
	incident-end	</desc></incident>		
add_quote **(citations)**	quote	<quote>texte</quote>	Utilisation d'un point de repère pour placer les balises sur le texte en question → guillemets **Attention :** certaines quote ne sont pas annotées car le point de repère est manquant dans l'OCR.	

	quote-beginning	<quote><seg>texte	**Attention :** traitement particulier pour les seg couplés avec quote-beginning et quote-end car chevauchement de balises. Pour ces cas, le seg est traité dans la fonction add_quote.	
	quote-end	texte</seg></quote>		

❖ *Script balisage logique*

	Étiquettes	Résultat XML TEI	Méthode et incertitudes	À ajouter
add_structure **(structure générale)**	body	<text> <body> texte	Faut-il inclure le texte ? Oui car besoin de récupérer numéro de page	**Ajouter :** @ana sur text
	body1	<text> <body> <pb n=”1”/>		**À inclure plus tard car traitement spécifique.**
	text	text	**Attention :** pas besoin	**À vérifier sur séances**

		 </body> </text>	d'inclure le texte avant les balises, car le texte de la boxe en question n'est pas à inclure (pied de page contenant l'information sur l'imprimerie).	**sans parties complémentaires ni annexes**
	back	texte </body> <back>		
	text-back	 </back> </text>	**Attention :** pas besoin d'inclure le texte avant les balises, car le texte de la boxe en question n'est pas à inclure (pied de page contenant l'information sur l'imprimerie).	
add_division **(divisions)**	part	 texte		**Ajouter plus tard :** @corresp
	part1	texte		
	agenda	 <head>texte</head>		

	appendices	 <div type=”appendices”> <head>texte</head>		
	part1-appendices	<div type=”appendices”> <head>texte</head>		**À vérifier sur séances sans parties complémentaires avant**
	erratum	 <head> <label>texte</label>		**À vérifier sur séance avec erratum**
	part1-erratum	 <head> <label>texte</label>		**À vérifier sur séances avec erratum**
	lists	 <head> <label>texte</label>		**À vérifier sur séances avec listes**
	part1-lists	 <head> <label>texte</label>		**À vérifier sur séances avec listes**
	offices	 <head>texte</head>		**À vérifier sur séances avec nomination des bureaux**

	part1-offices	 <head>texte</head>		**À vérifier sur séances avec nomination des bureaux**
	sitting		**Attention :** obligation de traiter sitting et contents en même temps car impossible d’appliquer 2 traitements différents sur une même boxe.	
	other-sitting	 <head>texte</head>		**À vérifier sur les CR contenant plusieurs séances**
	contents	 <head>texte</head> <list>	**Attention :** obligation de traiter sitting et contents en même temps car impossible d’appliquer 2 traitements différents sur une même boxe.	
	voting1	 <head> <label>texte</label>		**Ajouter :** 1ere div → @xml:id 2e div → @xml:id **Ajouter plus tard :** 2e div →@corresp

	voting	 <head> <label>texte</label>		**Ajouter :** @xml:id **Ajouter plus tard :** @corresp
	div-end			
	rectification	 <div type=”rectification”> <head>texte</head>		**Ajouter plus tard :** @corresp
	petition	 <head> <label>texte</label>		**À vérifier sur séance avec petition**
	part1-petition	 <head> <label>texte</label>		**À vérifier sur séance avec petition**
add_structural_comment **(commentaires structurels)**	note	<note> </note>		**Ajouter :** @xml:id
	voterslist-beginning	<note type=”voterslist”> <desc>texte</desc>		
	note-beginning	<note type="numbersannounced>texte		

	note-end	</note>		
add_item **(item)**	item	<item>text</item>		**Ajouter plus tard :** @xml:id
	item-list	<item>text</item> </list>		
add_title **(titres et titres divisés)**	head	<head>texte</head>		
	desc	<desc>texte</desc>		
	note-head	<note>texte</note> </head>		**Ajouter :** @xml:id
add_table **(tableau)**	table	<table> <row> <cell>Tout le texte du tableau</cell> </row> </table>		**À ajouter par la suite :** changer le balisage en fonction de l'analyse des tableaux à réaliser (garder <table> ou mettre <measure>)

❖ *Script compilation*

	Méthode	**Problèmes**
compilation	Appel de l'ensemble des fonctions permettant de baliser le contenu du CR.	

❖ *Script nettoyage*

	Méthode	Problèmes
clean_xml **(sauts de ligne)**	\n : à remplacer par un espace -\n : ne pas rajouter d'espace **À prendre en compte :** - Cas des mots divisés entre 2 boxes : → Si c dû à un changement de colonne alors il faut recoller le mot. → Si c dû à un changement de page, il faut garder le tiret car balise <pb> sera inclus dans le mot. → À appliquer directement sur les fichiers XML. **Attention :** - Cas de l'enchaînement de deux boxes : → Ajouter un espace, traitement à réaliser directement dans la boucle permettant d'écrire le texte dans le fichier XML.	**Défaut de la méthode** : quand le mot est réellement divisé en deux, et qu'il est sur deux lignes, il est recollé (exemple grand-père).
delete	Utilisation de l'étiquette "useless" afin de supprimer toutes les boxes que nous ne	

(suppression élément inutile)	souhaitons pas intégrer dans l'élément "text" (l'en-tête par exemple). Certaines boxes sont supprimées via le traitement de d'autres étiquettes directement (bandeau supérieur traité avec la gestion des changements de page, pied de page avec la gestion des étiquettes "text" et "text-back".	

❖ *Script métadonnées*

	Étiquette	Résultat XML TEI	Méthode et incertitudes	À ajouter
build_teiheader **(construction à la main du header complété par les variables issues de var_metadata)**		<date when="2022-07-25"/>	Utilisation de datetime : date.today().strftime("%Y-%m-%d")	**Ajouter :** une gestion automatique des éléments <tagsDecl> et <extent>
var_metadata **(variables contenant le texte utile pour les métadonnées issus**	date-pub	<date>27 novembre 1889</date> <publicationStmt>		**Ajouter :** @when

de data)	meeting-session	<meeting n="E1" ana="#parla.lower #parla.session">Session extraordinaire de 1889</meeting>	**Attention :** informations traitées dans une même condition car présentes sur la même ligne.	
	meeting-legislature	<meeting n="5L" ana="#parla.lower #parla.legislature">5e législature</meeting>		
	meeting-sitting	<meeting n="10" ana="#parla.lower #parla.sitting">10e séance</meeting>		
	date-sitting	<date>mardi 26 novembre<date>		

❖ *Script main*

	Méthode	**Problèmes**
Chemin relatif	Mettre un chemin relatif pour aller chercher les : - json (path_to_json) - xml (path_to_xml)	
Variables	Déclaration des variables utiles pour la création de l'élément racine :	

	- beginning_elements - end_elements	
Boucle générale	- Ouverture des fichiers JSON et .load() - Gestion des changements de page - Appel des métadonnées (fonction build_teiheader) - Appel des données à inclure dans l'élément "text" (fonction compilation) - Sous-boucle permettant de créer un fichier XML pour chaque séance, que l'on nomme selon le nom du fichier json, dans lequel on écrit l'ensemble des éléments et pour lequel on nettoie les espace	
Gestion des inclusions dans le fichier corpus	Le fichier corpus doit être situé dans le dossier xml_data. Utilisation de la librairie lxml pour parser le fichier et inclusion automatique des éléments xi:include.	**Défaut :** si l'on supprime un fichier XML composant, alors la suppression de l'inclusion doit se faire à la main.
Amélioration des scripts : éléments à traiter		
Tests unitaires		
Vérifier la validité de l'XML	**Solution provisoire :** la validation du	**Solution à long terme :** reprendre le

	schema se fait manuellement pour l'instant. Le schema est précisé dans chaque en-tête des fichier XML, il faut ouvrir un éditeur XML et voir si la validation fonctionne. **Solution à long terme :** utilisation de la librairie lxml pour valider le schéma RNG.	code présent à la fin du fichier main et le modifier, car il ne fonctionne pas.
Gestion différents OS	Utiliser le module pathlib : - https://docs.python.org/3/library/pathlib.html - https://www.digitalocean.com/community/tutorials/how-to-use-the-pathlib-module-to-manipulate-filesystem-paths-in-python-3-fr	
Vérification étiquettes	Script à intégrer dans la chaîne de traitement.	
Gestion de certaines métadonnées du fichier corpus	Mettre en place une construction automatique pour : - l'élément <tagsDecl> - l'élément <particDesc> - l'élément <standOff>.	