Ingénierie documentaire

Projet

Prof. Dr.-Ing. Michael Piotrowski[1]

[1] Section des sciences du langage et de l'information · Université de Lausanne

Slide 2

Projet

Téléchargez le texte source : Cours Moodle → XML → `martin_clean.txt`

1. Analysez la structure et identifiez les éléments sémantiques
2. Définissez le format cible sur la base des recommandations de la TEI
3. Identifiez les régularités dans le texte source pour formuler des regex

Au cours de cette session et de la suivante, vous travaillerez sur un projet. Je vous encourage à collaborer en groupes de trois ou quatre, mais vous devez effectuer toutes les étapes vous-même. L'objectif du projet sera de transformer la version texte de la comédie *La tête de Martin*[1] en un document XML et, au cours de la session suivante, d'écrire une feuille de style XSLT pour transformer le document XML en un document HTML à afficher.

Les informations ci-dessous ont pour but de vous familiariser avec le texte et de vous préparer au projet.

1 Analyse

Vous trouvez sur Moodle dans le dossier *XML* la version originale (`martin.txt`) ainsi qu'une version « nettoyée » sans le texte de la licence et les pages liminaires (`martin_clean.txt`).

Voilà le début du texte :

```
SCÈNE I

BERTRAND, _seul_. (_Il est assis devant une table à droite_).

Maintenant, voyons si l'on a bien inscrit tous les voyageurs... (_Il
ouvre un registre_). M, Dubois, très bien; M. Lefèvre; M. Coquelet, très
bien; au numéro 9, M. Martin, profession, propriétaire; au numéro 11, M.
Martin... Tiens, encore un Martin! profession: professeur de prothèse
dentaire; au numéro 13, M. Martin!... Ah! çà, il n'y a donc que des
Martin cette année?... profession: clerc de notaire et célibataire!...
```

1. Mise à disposition par Project Gutenberg : http://www.gutenberg.org/cache/epub/12603/pg12603.txt

```
Ah! je le connais, celui-là... c'est le casse-cou qui est ici depuis un
mois.


SCÈNE II

BERTRAND, DURAND, puis VENCESLAS.


DURAND (_du seuil de la porte_).

Pardon, monsieur, n'auriez-vous pas ici un nommé Martin?

BERTRAND.

Oui, Monsieur; j'en ai même plusieurs.
```

Votre **première tâche** (en groupes) sera d'analyser la structure et d'identifier les éléments sémantiques.

2 Format cible

Un véritable projet de recherche ou de publication produirait idéalement un document XML conforme aux recommandations de la Text Encoding Initiative (TEI), notamment ceux concernant les « textes performatifs » décrits dans le chapitre 7 « Performance Texts ». Ce n'est pas nécessaire pour ce projet, mais nous allons nous inspirer de ce modèle ; il facilite aussi une transformation ultérieure dans le format standard.

Votre **deuxième tâche** sera de définir le format cible sur la base des recommandations de la TEI. Transformez un extrait court à la main pour créer un exemple.

3 Approche

Troisième tâche : identifiez les *patterns* dans le texte source que vous pouvez potentiellement utiliser pour la formulation d'expressions régulières.

Notez que le format de la source n'est pas complètement régulier. C'est pourquoi une approche interactive et étape par étape sera nécessaire. Cependant, il devrait être possible d'utiliser des expressions régulières pour effectuer la plus grande partie de la transformation.

4 Remarques

Pendant le travail, prenez des notes. En particulier, notez les expressions régulières que vous avez utilisé.

Sauvegardez la version du texte avant et après chaque étape.

Il est parfois nécessaire d'insérer des balises *autour* d'éléments textuels étendus, par exemple `<sp>` et `</sp>` autour des dialogues. Plutôt que de tenter de construire une expression régulière complexe pour capturer un nombre indéfini de lignes, il est souvent beaucoup plus simple de procéder en plusieurs étapes. Par exemple, il est facile d'identifier avec une expression régulière les positions où il faut insérer la balise ouvrante `<sp>` (ainsi que `<speaker>` et `<p>`), indiquées par ➡ dans l'extrait suivant. Par contre, il n'est pas du tout trivial de capturer le texte du dialogue au même temps, ce qui nécessite d'identifier les positions indiquées par ⬅.

```
➡BERTRAND.
```

```
Oui, Monsieur; j'en ai même plusieurs.⬅

➡DURAND.

Plusieurs Martin valent mieux qu'un. (_À la cantonade_.) Viens,
Venceslas.⬅

➡BERTRAND.

Monsieur désire une chambre?⬅
```

Cependant, si l'on a d'abord transformé le texte en :

```
<sp><speaker>BERTRAND</speaker><p>

Oui, Monsieur; j'en ai même plusieurs.⬅

<sp><speaker>DURAND</speaker><p>

Plusieurs Martin valent mieux qu'un. (_À la cantonade_.) Viens,
Venceslas.⬅

<sp><speaker>BERTRAND</speaker><p>

Monsieur désire une chambre?
```

il est ensuite facile d'insérer les balises fermantes aux positions indiquées par ⬅ : on peu utiliser les balises ouvrantes pour identifier la fin du dialogue précédent et après il faut juste insérer la dernière manuellement. En fait, on pourrait même directement insérer `</sp><sp><speaker>`… et après enlever la première balise `</sp>` (et en ajouter une à la fin).

Quand on n'a pas besoin de *déplacer* du texte, ce type d'approche est en général plus efficient que de tenter de concevoir une expression régulière complexe avec des groupes (en particulier s'il nécessiterait de capturer plusieurs lignes).