Documentation de l'encodage réalisé dans le cadre de la validation du séminaire XML-TEI du Master HN

Alexandre Lionnet-Rollin

alexandre.lionnet@chartes.psl.eu

13 Mai 2024

Introduction

Dans le cadre de ce rendu pour le séminaire de structuration de données, j'ai choisi d'encoder un des textes que j'étudie dans le cadre de mes recherches, la creepypasta *Candle Cove*[1].

Les creepypastas sont des courtes productions écrites, nativement numérique, qui ont une double vocation : être virale, et faire peur. Ces petites histoires, souvent anglophones, sont à rapprocher d'autres formes de littérature ou de productions écrites ou orales comme les histoires folkloriques ou bien la lit-

[1] Il est possible de consulter l'histoire ici

térature fantastique.

Le choix de cette histoire en particulier est motivé par son statut dans l'histoire de ce "genre" : c'est une des premières et des plus connues aujourd'hui. Ayant sûrement servi d'exemple à d'autres histoires, le but de l'encodage est de mettre en valeur à la fois la structure particulière de cette histoire (sous la forme de publications sur un forum fictif), et les thèmes récurrents caractéristiques.

1 L'encodage

Afin de rendre compte de la forme de la publication originale, j'ai décidé d'utiliser les balises `<div>` afin de marquer chaque réponses des utilisateurs. Ces balises portent deux éléments : un `type` et un `xml:id` pour préciser qu'il s'agit respectivement d'un "post" puis le numéro du post. Voici un exemple:

```
<div type="post" xml:id="post_1">
```

Concernant la structure des post, chacun suit une forme similaire qu'on cherche ici à conserver : le nom de l'utilisateur et le sujet du post. Afin de rester au plus proche de la publication originale j'ai ajouté l'attribut `rend="bold"` pour marquer la différence entre le corps du texte et le reste. Exemple:

```
<head rend='bold'>Skyshale033</head><lb/>
<head rend='bold'>Subject: Candle Cove local kid's show?</head><lb/>
```

Dans l'histoire, différents personnages fictifs ou non sont mentionnés. Pour en rendre compte, j'ai d'une part identifier et encoder tous les personnages dans le `teiHeader` plus particulièrement dans le `particDesc`, en distinguant les deux types de personnages :

```
<particDesc>
        <listPerson type='users'>
            <person xml:id="mike_painter65">
                <persName>mike_painter65</persName>
            </person>
            <person xml:id="Skyshale033">
                <persName>Skyshale033</persName>
            </person>
            <person xml:id="Jaren_2005">
                <persName>Jaren_2005</persName>
            </person>
            <person xml:id="kevin_hart">
                <persName>kevin_hart</persName>
            </person>
        </listPerson>
        <listPerson type='fictional'>
            <person xml:id="Janice">
                <persName>Janice</persName>
            </person>
            <person xml:id="Pirate_Percy">
                <persName>Pirate Percy</persName>
            </person>
            <person xml:id="Skin_Taker">
                <persName>The Skin-Taker</persName>
            </person>
            <person xml:id="Horace_Horrible">
                <persName> Horace Horrible</persName>
```

On notera alors toutes les mentions par les utilisateurs de ces personnages

avec la balise `<persName>` et l'attribut `ref` qui aura pour valeur l'`xml:id` défini dans le `teiHeader`. Exemple :

```
<persName ref="#Pirate_Percy">Pirate Percy</persName>
```

et

```
<persName ref="#Skyshale033">Skyshale</persName>
```

Je n'ai pas suivi la même méthode néanmoins pour les noms utilisateurs en en-tête de chaque post : cet encodage semble superfétatoire et risque d'alourdir l'encodage global.

Certains éléments peuvent être inconnus pour le lecteur, particulièrement s'il n'est pas Américain dans notre cas. Pour ce faire, j'ai utilisé la balise `note` pour ajouter des éléments contextuels concernant les lieux mentionnées et certains éléments. Exemple:

```
I lived in <placeName>Ironton</placeName>
<note>Village de l'Ohio, USA</note> at the time.
```

Le coeur de l'encodage réside dans l'encodage des sujets (ou *topics*) présents dans ce texte. Pour ce faire j'ai utilisé un *topic modelling* réalisé sur l'ensemble de mon corpus pour déterminer quels *topics* souligner[2]. Pour encoder les *topics*, j'ai commencé par les lister dans le `teiHeader` (avec un `xml:id` unique et une courte description:

[2] voir en Annexe pour le résultat du *topic modelling*

```
<list xml:id='topics'>
    <item xml:id="topic_peur">
        <term type="topic">Peur, frayeur et autres</term>
    </item>
```

Puis j'ai noté chaque occurrence de chaque *topic* grâce une balise `<seg>` avec l'attribut `type="topic"` et `corresp="nom_du_topic"`. Exemple :

```
i <seg type="topic" corresp="#topic_souvenir">remember</seg>
seeing what you described.
they just <seg type="topic" corresp="#topic_son">screamed</seg>
```

Enfin il convient de noter que le texte présente des fautes d'orthographes: j'ai décidé de les encoder avec la balise `<seg>` avec l'attribut `type="erreur"` afin de simplement les signaler. J'ai décidé de les garder afin de maintenir ce style assez typique des forums internet, où le plus important est de réagir, quitte à ne pas faire attention à son orthographe. Exemple:

```
his jaw just slid back and <seg type="erreur">foth</seg>
```

Dans les deux derniers cas, l'utilisation de la balise `<seg>` est un choix avant tout pratique: cette balise n'est restrictive dans son utilisation et permet une grande flexibilité, sans contraindre le rendu final.

2 Les requêtes XPath

Voici quelques requêtes `XPath` afin de parcourir le document tout en permettant de vérifier la conformité de l'encodage:

- Vérifie l'absence de `#` devant la valeur de `corresp` dans les segments encodant les topics (on espère que la requête ne retourne rien)
  `//seg[@type='topic'][not(starts-with(@corresp,'#'))]`
- Permet d'afficher tous les posts
  `//p[not(ancestor::teiHeader) and parent::div[type='post']]`
- Permet de faire apparaître les contenus du topic lié à l'enfance
  `//seg[type='topic' and @corresp='#topic_enfance']`
- Permet de faire apparaître les fautes encodées
  `//seg[@type="erreur"]`
- Permet de sélectionner les post de l'utilisateur Skyshale033
  `//div[head[contains(.,'Skyshale033')]]/p`
- Permet de faire apparaître les publication qui font mention à la fois de la peur et de son
  `//p[.//seg[corresp='#topic_peur'] and .//seg[corresp="#topic_son"]]`

3 Transformation XSLT

Pour la feuille de transformation XSLT, j'ai choisi de produire une transformation au format HTML. En plus de garder une structure balisée, le langage HTML, couplé avec le langage CSS, permet un affichage plus riche du texte. Ainsi, le résultat de la transformation est une page HTML où :

- La forme générale est conservée (le titre, la division en post séparés avec l'en-tête en gras et les saut de lignes)
- Chaque occurrence d'un topic est colorée d'une couleur différente : cela permet de visualiser plus clairement les topics, et de voir les co-occurrences des thèmes de façons plus claire
- Les notes sont gardées et transformées en note de bas de page, avec des renvois vers le bas de la page au niveau des mots.
- Les fautes d'orthographes sont conservées et soulignées pour les marquer.

Je vous renvois à la feuille de transformation pour plus de détails.

4 Limites et potentiels

Dans le cadre de mes recherches, j'ai choisi de me concentrer sur une analyse quantitative des creepypastas, en utilisant une quantité substantielle de données. À titre d'exemple, l'une des plateformes que j'examine renferme pas moins de 200 000 publications. Cette échelle importante souligne la nécessité de trouver un équilibre entre la richesse de l'encodage et sa faisabilité pratique.

Bien que la priorité actuelle soit de traiter efficacement ce volume considérable de données, je reconnais le potentiel d'enrichir davantage l'encodage pour capturer des aspects plus subtils et nuancés des textes. Par exemple, il

pourrait être intéressant d'incorporer des marqueurs stylistiques spécifiques que l'on retrouve fréquemment dans ce genre de narration, tels que des motifs récurrents, des figures de style particulières ou des techniques d'écriture distinctives.

Cependant, cette entreprise nécessiterait une exploration plus approfondie de mes recherches et une compréhension plus fine des structures narratives et stylistiques des creepypastas. À mesure que mes travaux progressent, je suis ouvert à l'idée d'affiner et d'élargir l'encodage pour mieux saisir la complexité et la diversité de ces récits contemporains d'horreur en ligne.

5 Annexe

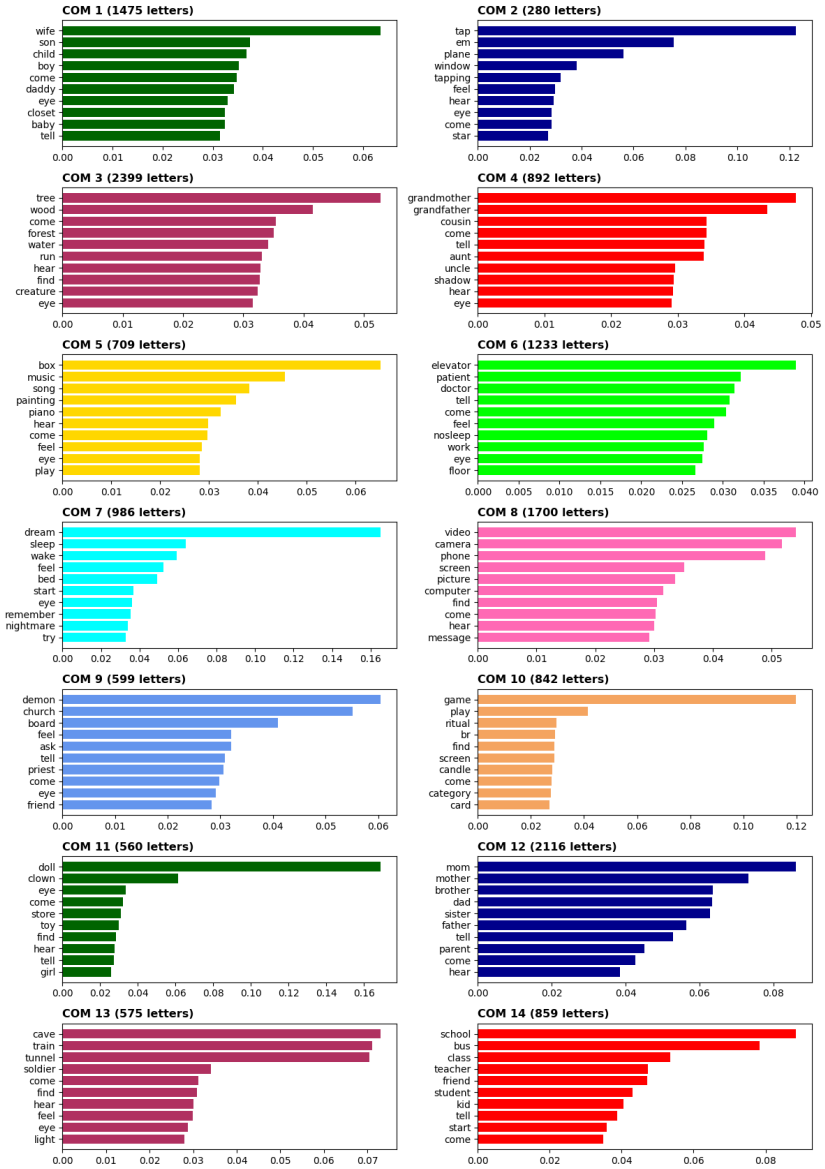

Figure 1: Topic modelling réalisé grâce à un tf-idf sur un échantillon représentatif de mon corpus