XQuery pour les Humanités numériques

par Farid Djaïdja
sous la direction de Xavier-Laurent Salvador et Fabrice Issac

Ce mémoire de M2 a été raccourci afin de proposer une version prête à l'usage à l'attention des chercheurs désireux de se former au XML et à XQuery. La numérotation est celle d'origine.

Table des matières

À Joël, Xavier-Laurent, Fabrice, Ibtissem et Svitlana.

Sursum Corda.

L'humanisme est un courant culturel européen, trouvant ses origines en Italie, principalement autour de Florence, qui s'est développé à la Renaissance. Renouant avec la civilisation gréco-romaine, les intellectuels de l'époque manifestent un vif appétit de savoir (philologie notamment). Considérant que l'Homme est en possession de capacités intellectuelles potentiellement illimitées, ils considèrent la quête du savoir et la maîtrise des diverses disciplines comme nécessaires au bon usage de ces facultés. Ils prônent la vulgarisation de tous les savoirs.

Wikipédia

Le texte est au cœur des Humanités. Si la question semble ne pas se poser, c'est que l'on tient sa définition pour acquise. Mais qu'est-ce qu'un texte ? Si l'on considère des mots couchés sur le papier, c'est se méprendre sur l'objet. Avant de prendre forme, un texte passe par plusieurs étapes depuis sa conception jusqu'à sa lecture. Si le texte papier passe par une mise en page, avec un choix de police de caractères, il est pensé en amont à travers sa structure, intrinsèquement liée au contenu et donc à la donnée, et son inscription dans un courant. Aujourd'hui, se posent pour le texte numérique en plus les questions de l'encodage, qui est la première abstraction du texte numérique, du format, de la numerisation, toutes ces étapes qui font qu'un texte existe, sur le papier ou à l'écran.

De telles considérations peuvent échapper à l'Humaniste numérique, qui est parfois amené à sous-estimer cette dimension, souvent par méconnaissance de cet objet textuel. La linguistique a déjà bénéficié de sa mue à travers l'émergence de la linguistique informatique et du traitement automatique de la langue (TAL). Toutefois, le texte est souvent traité comme un simple flux de données, ignorant les spécificités de la langue et du texte, notamment cette structure.

L'ambition de ce mémoire est de présenter un état de la technique à propos du traitement automatique de la langue, afin de mettre en évidence les apport qu'XML et XQuery peuvent faire valoir dans une meilleure appréhension du texte numérique, à la fois au niveau des unités lexicales, du fragment et du corpus, afin que l'Humaniste numérique puisse bénéficier d'une autonomie par rapport à l'outil informatique et en exploiter tout sa puissance.

Enfin, il brosse des perspectives de recherches, à la fois sur le plan théorique et applicatif, des résultats que l'on peut obtenir lorsque l'on a la maîtrise à la fois sur la donnée et sur l'information, et revenir aux origines du texte : son sens.

2. XML, un formalisme orienté vers la recherche.

Les outils actuels du Traitement automatique des langues permettent d'explorer le texte uniquement dans sa structure de surface. Mais plus qu'un ensemble de mots couchés sur cette surface, un texte possède une structure profonde et un formalisme. La ponctuation est un outil qui permet d'aborder la structuration d'un texte, car elle est formelle et univoque : elle ne permet pas d'ambiguïté. Le XML, pour *eXtensible Markup Language*, est un formalisme qui permet d'ajouter aux mots une couche d'abstraction et de baliser un texte de manière invisible et libre. Le W3C (*World Wide Web Consortium*), l'organisme de normalisation du web, définit le XML comme « un format de texte simple dérivé du SGML (ISO 8879) »[1], tout comme le HTML (*HyperText Markup Language*).

Le SGML, ou *Standard Generalized Markup Language*, est né en 1969 des laboratoires d'IBM comme un métalangage de balisage conçu pour faciliter l'échange et la révision de notices. Il instaure un système de *tags* ou balises pour structurer la donnée textuelle en arborescence[2]. Le HTML apparaît en 1991 sous la houlette du W3C afin de répondre aux besoins du web. Il permet la construction de pages web en suivant un formalisme mêlant métadonnées et données respectivement contenues dans les balises `<head>` et `<body>` qui est interprété côté client par un navigateur web comme Internet Explorer, Mozilla Firefox ou Google Chrome.

La philosophie du XML aujourd'hui est d'autoriser tout ce qui n'est pas interdit en visant à offrir l'éventail de souplesse le plus large possible avec un mélange de liberté et de rigueur, à l'opposé de la philosophie des applications en HTML, plus permissives dans leur formalisme. Ces deux pôles sont les extrêmes de l'axe XML. Il est en effet possible de moduler l'emploi que l'on veut faire du XML pour tendre davantage vers un formalisme ouvert ou verrouillé.

[1] http://www.w3.org/XML/.
[2] http://xmlfr.org/documentations/articles/000321-0001.

XML permet de superposer plusieurs couches d'abstraction sur le document. La première se situe au plus proche du texte : les balises permettent de créer une zone de texte que l'on peut enrichir d'informations linguistiques ou métalinguistiques.

2.1. Une cartographie du texte : les balises comme stratification des lieux de discours

2.1.1. Un marquage des lieux de discours: les balises

Le HTML est un dialecte du XML. Ils partagent la même syntaxe, mais le HTML impose une nomenclature pour permettre une interprétation via un navigateur. Le HTML et le XML se distinguent par leur philosophie mais aussi par leurs applications. Le développement en HTML est destiné à l'affichage de texte. Il se trouve à la fois plus contraignant et plus laxiste car une page HTML doit répondre à un cahier des charges qui lui est propre et à un but fixé. Le lexique utilisé n'est pas libre. L'emploi de certaines balises est imposé, par exemple `<a>` est réservé pour les liens hypertextuels et pour rien d'autre en HTML[3].

À l'inverse, le XML est plus souple car aucune nomenclature n'est nativement imposée et il se fonde avant tout sur un système de balises. Il existe néanmoins quelques vives recommandations, telles que l'emploi de l'UTF-8, qui est la table Unicode recommandée dans une volonté de standardisation de l'encodage[4]. Tout comme en HTML, ce système est composé de balises permettant d'encadrer l'information, de manière invisible. Les balises constituent la structure du texte, l'enrichissant de métadonnées. Ces métadonnées ne sont pas au même niveau que les données :

> Unijambiste, **<personnage>**Cyrus**</personnage>** redevenait lui-même. **<personnage>**Adam**</personnage>** retrouvait le père de son enfance. Au début de leur rencontre, il avait ressenti un léger mépris, mais maintenant, la peur, le respect, l'animosité de sa jeunesse renaissaient en lui,

[3] Voir Annexe 1.
[4] Voir pour plus d'informations http://www.unicode.org/standard/translations/french.html.

> et il se retrouvait petit garçon à deviner l'état d'esprit de son père pour éviter tout ennui.[5]

Ce système de balises permet d'encadrer l'information de manière invisible. En effet, l'intérêt même d'un document XML est de pouvoir enrichir un texte sans que cela ne soit un frein à sa lisibilité. Dans l'exemple ci-dessus, la balise `<personnage>` permet de cibler l'information, mais également de la délimiter en indiquant son début et sa fin. Ainsi, *Cyrus* sera enrichi de cette information tout comme *Adam* et celle-ci pourra être exploitée par la suite. Ce double niveau d'information peut être observé sur le web, où des pages qui apparaissent comme de simples textes sont en fait enrichis de nombreuses informations.

Les balises vont toujours par paire, elles peuvent être soit ouvrantes, `<personnage>`, soit fermantes comme `</personnage>`. Le formalisme XML impose la présence d'une balise fermante pour chaque balise ouvrante. Un document XML ne respectant pas cette syntaxe sera dit mal formé. Les navigateurs peuvent s'affranchir de la bonne formation d'un document et permettre malgré tout un affichage. Ce laxisme a conduit à rendre la production de documents HTML moins rigoureuse et un nombre non négligeable de pages sur le web s'avèrent être mal formées. Une telle chose est impossible pour un document XML, le respect de cette syntaxe est le premier niveau du formalisme. Le jeu de balises ouvrantes et fermantes constitue ce que nous qualifions de syntaxe horizontale, une balise ouvrante précède obligatoirement une balise fermante. Celle-ci se situe donc au niveau de la balise. En plus de celle-ci, il existe une autre syntaxe, dite verticale.

Un document XML est un fichier contenant des données organisées et structurées. C'est à ce titre qu'ils sont exploités comme base de données. Si une base de données SQL à la représentation d'un tableau Excel, un document XML possède une arborescence. Et à la base de cet arbre, on trouvera une racine. La racine d'un document XML est un couple de balise qui va englober tout le document, données comme balises. Il ne peut et ne doit exister qu'une racine par document XML. Partant de la racine, un document XML peut être considéré comme des poupées russes où chaque poupée est une balise contenant d'autres balises. La structure du *Cid* pourrait être représentée de la façon suivante en XML :

```
<racine>
  <titre>Le Cid</titre>
```

[5] John Steinbeck, *À l'est d'Eden*.

```
<acte1>
  <scene1>Elvire et Chimène</scene1>
  <scene2>Doña Urraque amoureuse de Rodrigue</scene2>
  <scene3>Don Diègue au poste de gouverneur</scene3>
  <scene4>Monologue de Don Diègue</scene4>
  <scene5>Volonté de tuer Don Gomès</scene5>
  <scene6>Monologue de Rodrigue</scene6>
</acte1>

<acte2>
  <scene1>Don Arias et Don Gomès</scene1>
  <scene2>Duel entre Don Gomès et Don Rodrigue</scene2>
  <scene3>Discussion entre l'Infante et Chimène</scene3>
  <scene4>Annonce du duel en cours</scene4>
  <scene5>Discussion entre l'infante et Léonor</scene5>
  <scene6>Le refus de Don Gomès</scene6>
  <scene7>La mort de Don Gomès</scene7>
  <scene8>Chimène demande vengeance</scene8>
</acte2>

<acte3>
  <scene1>Rodrigue vient chez Chimène</scene1>
  <scene2>Chimène arrive accompagnée de Don
          Sanche</scene2>
  <scene3>Les sentiments de Chimène</scene3>
  <scene4>Rodrigue vient offrir sa vie à Chimène</scene4>
  <scene5>Monologue de Don Diègue</scene5>
  <scene6>L'honneur plus important que l'amour</scene6>
</acte3>

<acte4>
  <scene1>Rodrigue victorieux</scene1>
  <scene2>L'Infante vient prendre part aux douleurs de
          Chimène</scene2>
  <scene3>Les félicitations de Rodrigue</scene3>
  <scene4>Arrivée de Chimène</scene4>
  <scene5>Chimène encore amoureuse de Rodrigue demande un
          combat qui tranche le différend</scene5>
</acte4>

<acte5>
  <scene1>Rodrigue offre une nouvelle fois sa vie à
          Chimène</scene1>
  <scene2>Monologue de l'infante</scene2>
  <scene3>Léonor vient réconforter l'Infante</scene3>
  <scene4>Désarroi de Chimène</scene4>
  <scene5>La méprise de Chimène</scene5>
  <scene6>Le roi détrompe Chimène</scene6>
  <scene7>Le mariage est résolu</scene7>
</acte5>
```

```
</racine>
```

Ce document est bien formé : il possède une racine unique et chaque balise ouvrante appelle une balise fermante. La profondeur de l'arborescence y est illustrée. Un autre élément qui participe à la bonne formation du document est le non chevauchement des balises.

<acte1> *<scene1>* Elvire et Chimène *</scene1>* **<acte1>** est bien formé.

<acte1> *<scene1>* Elvire et Chimène **</acte1>** *</scene1>* est mal formé.

Remise dans un schéma en arborescence, cette mauvaise formation est encore plus saillante :

```
<racine>
                <acte1>

                <scene1> Elvire et Chimène
                </acte1>

                </scene1>
</racine>
```

Un document XML est un arbre généalogique où chaque élément se définit par rapport à ses ascendants, ses descendants et par les autres membres de la même génération. Chaque élément a donc son identité propre dans l'arbre XML, mais il est possible de les caractériser davantage grâce à des attributs.

2.1.2. La spécification des nœuds de l'arborescence : les attributs

Les attributs permettent d'apporter des informations aux balises. Les balises connaissent des contraintes dans le choix des caractères. En effet, il est recommandé de se limiter à ceux de la table ASCII pour éviter des problèmes d'encodage. Ainsi, sont à proscrire tous les caractères avec des marques diacritiques. La valeur des attributs est entre guillemets.

À ce titre, il s'agit d'une chaîne de caractères et l'emploi de diacritiques n'est plus prohibé. Dans l'exemple du balisage du *Cid*, nous pouvons envisager l'utilisation des balises et attributs suivants :

```
<racine>

  <titre>Le Cid</titre>

  <acte id="1">
    <scene id="1">Elvire et Chimène</scene>
    <scene id="2">Doña Urraque amoureuse de Rodrigue</scene>
    <scene id="3">Don Diègue au poste de gouverneur</scene>
    <scene id="4">Monologue de Don Diègue</scene>
    <scene id="5">Volonté de tuer Don Gomès</scene>
    <scene id="6">Monologue de Rodrigue</scene>
  </acte>

  <acte id="2">
    <scene id="1">Don Arias et Don Gomès</scene>
    <scene id="2">Duel entre Don Gomès et Don
                  Rodrigue</scene>
    <scene id="3">Discussion entre l'Infante et
                  Chimène</scene>
    <scene id="4">Annonce du duel en cours</scene>
    <scene id="5">Discussion entre l'infante et
                  Léonor</scene>
    <scene id="6">Le refus de Don Gomès</scene>
    <scene id="7">La mort de Don Gomès</scene>
    <scene id="8">Chimène demande vengeance</scene>
  </acte>

</racine>
```

En se servant des attributs, on ne multiplie plus les balises qui étaient alors à usage unique mais on établit une classe qui est caractérisée. La pièce n'est plus segmentée en un `acte1`, un `acte2` jusqu'à un `acte5`, mais en `acte`. Chaque acte se voit affecter un identifiant. Ainsi, le premier acte sera un `acte` parmi d'autres `acte`, à la différence que son identifiant sera `1`. Cette caractérisation est plus proche ontologiquement de la représentation que l'on se fait de la structure d'une pièce de théâtre, mais elle facilitera aussi la recherche par la suite. La caractérisation ne se limite pas à un identifiant mais peut aller plus loin grâce à la multiplication des balises. Un conjugueur pourrait se baser sur la base de données XML suivante :

```
<racine groupe="1er" verbe="manger">
```

```
    <conjugaison mode=”indicatif” temps=”présent”>
      <personne id=”1” nombre=”singulier”>mange</personne>
      <personne id=”2” nombre=”singulier”>manges</personne>
      <personne id=”3” nombre=”singulier”>   mange</personne>
      <personne id=”1” nombre=”pluriel”>mangeons</personne>
      <personne id=”2” nombre=”pluriel”>mangez</personne>
      <personne id=”3” nombre=”pluriel”>mangent</personne>
    </conjugaison>

    <conjugaison mode=”subjonctif” temps=”imparfait”>
      <personne id=”1” nombre=”singulier”>mangeasse</personne>
      <personne id=”2” nombre=”singulier”>mangeasses</personne>
      <personne id=”3” nombre=”singulier”> mangeât</personne>
      <personne id=”1” nombre=”pluriel”>mangeassions</personne>
      <personne id=”2” nombre=”pluriel”>mangeassiez</personne>
      <personne id=”3” nombre=”pluriel”>mangeassent</personne>
    </conjugaison>

  </racine>
```

En remontant l’arborescence de la donnée à la racine, « mangeassions » correspond à la première personne du pluriel au subjonctif imparfait du verbe du premier groupe « manger ». XML offre une très grande liberté de choix dans la construction de la structure de données. Cette flexibilité est un avantage certain, mais peut également s’avérer être source de problème, en particulier dans le cadre d’un travail collaboratif. La structure doit être la plus uniforme possible pour pouvoir être exploitée au mieux, c’est pourquoi il est parfois nécessaire d'apposer une couche supplémentaire de formalisme afin de garantir une homogénéité.

2.2. La DTD, un degré supplémentaire dans le formalisme

Une harmonisation passe par l’établissement de règles communes. L’un des écueils du travail d’équipe est l’évidence. Considérer une nomenclature comme une évidence est une gageure. Ce qui peut paraître évident pour soi ne l’est pas nécessairement pour autrui car chaque grille de lecture est la manifestation d’une subjectivité. XML est un monde ouvert

avec très peu de contraintes, il est donc aisé de tomber dans une anarchie formelle. Cette anarchie est néfaste à l'optimisation d'une base de données car elle a des répercussions sur son temps de réponse et donc sur ses performances. Elle nuit également à sa lisibilité et rendra la rédaction moins aisée. C'est pourquoi il est parfois nécessaire d'adosser à un document XML un *Document Type Definition* (DTD) qui définira le champ des possibles dans l'utilisation des balises et des attributs. La bonne formation d'un document XML constitue le premier contrôle, la validité par rapport à une DTD sera le second.

HTML a été défini plus haut comme un dialecte de XML. Il s'agit davantage d'un XML doté d'une DTD bien particulière, où seules certaines balises sont autorisées, où certaines balises peuvent avoir un attribut particulier et d'autres doivent absolument en avoir un bien précis. Cette DTD garantie une harmonisation des pages à destination du web afin qu'elles soient interprétées correctement par les navigateurs. De la même manière, il est possible de créer sa propre DTD qui pourra être consultable comme document de référence pour l'ensemble des rédacteurs du fichier XML. La rédaction d'une DTD contrôle de manière absolue le choix de structuration des rédacteurs, elle doit donc pouvoir subvenir à tous les besoins en les anticipant. Ce contrôle se fait sur six niveaux : le nom des balises, le contenu des balises, leur nombre, le nom des attributs, la valeur des attributs et la nécessité de leur présence.

2.2.1. Le nom des balises

Une DTD possède deux types de déclaration : les déclarations de balises et celles d'attributs[6]. Les balises se déclarent de la manière suivante :

```
<!ELEMENT racine>
```

L'écriture d'une DTD doit se faire avec un soin particulier, car elle sera extrêmement contraignante. L'orthographe n'admet pas d'approximation. Toute balise ne correspondant pas à `<racine>` sera rejetée lors de la validation. Seront rejetés `<racines>`, `<Racine>`, `<RACINE>` car la casse n'est pas ignorée.

[6] Voir Annexe 2 pour un exemple complet de DTD.

2.2.2. Le contenu des balises

Une balise peut contenir deux types d'information : des données ou d'autres balises. La donnée sera textuelle, sous forme de chaîne de caractère. Le texte est le degré le plus bas de l'arborescence XML. Le but de cette arborescence est précisément de structurer cette donnée. Au niveau des balises, la donnée textuelle est déclarée sous le code `#PCDATA`. Les balises déclarées comme descendants sont simplement nommées.

```
<!ELEMENT racine (titre|acte|#PCDATA)>
```

Dans cet exemple, la racine du document XML pourra être composée de trois éléments uniquement : une balise `<titre>`, une balise `<acte>` ou du texte. Cette liste est exhaustive mais sans contraindre à l'emploi obligatoire des trois.

2.2.3. Le nombre de balises

La DTD contraint à la nature des descendants d'une balise, mais également leur nombre. Cette précision se fait à l'aide d'un emprunt aux expressions rationnelles[7]. Ainsi, il sera possible d'adjoindre des multiplicateurs aux déclarations de descendance :

- `?` indique que la balise descendante pourra être omise, mais si elle est présente, elle ne pourra l'être qu'une fois ;
- `+` contraint à ce que la balise soit présente au moins une fois sans limiter son nombre d'occurrences ;
- `*` laisse la possibilité d'employer la balise ou pas, et si elle est employée elle ne sera pas limitée en nombre d'occurrences.

[7] Pour plus de précisions sur les expressions rationnelles, J. Friedl. *Mastering regular expressions*. Sebastopol, CA : O'Reilly Media, Inc., 2006.

```
<!ELEMENT racine (titre|acte|#PCDATA)+>
```

Dans cet exemple, la balise <racine> doit avoir pour descendant obligatoirement au moins une balise <titre>, au moins une balise <acte> et obligatoirement du texte hors balise.

2.2.4. Le nom des attributs et leur valeur

Tout comme pour les balises, le nom des attributs doit pouvoir être contraint si la rédaction du document XML exige un haut niveau de rigueur et d'homogénéité. La déclaration des attributs possibles se fait en les rattachant à une balise. La valeur de l'attribut doit également être précisée pour chacun d'entre eux, à savoir si la valeur est imposée ou libre. Si elle est libre, le code employé sera CDATA. La valeur de l'attribut étant une chaîne de caractères, l'emploi des diacritiques est possible. Ceux-ci ne seront alors pas optionnels à la rédaction mais obligatoires. Si la DTD indique comme valeur d'attribut possible « métaphore » et que la valeur des attributs est imposée, employer « metaphore » rendra le document invalide vis-à-vis de la DTD. La déclaration se fait par balise, il est possible de déclarer plusieurs attributs ainsi que leurs valeurs possibles :

```
<!ATTLIST scene moment (jour|nuit)
          decor (intérieur|extérieur)
          comediens CDATA>
```

Pour une scène, la DTD impose que pour chaque balise scene la valeur de l'attribut moment ne puisse être que « jour» ou « nuit» et aucun autre, et que celle de l'attribut decor ne puisse être que « intérieur » ou « extérieur ». En revanche, pour comediens, la valeur de cet attribut est libre.

2.2.5. L'optionalité des attributs

La DTD est un document devant parer à toutes les éventualités. Toutefois, tous les éléments de la structure ne sont pas à un même niveau d'importance. Dans l'exemple précédent, indiquer les éléments de mise-en-scène pour chaque scène peut sembler important, du moins plus important que le nom des comédiens. La DTD arrête quels sont les attributs obligatoires et lesquels sont optionnels à l'aide des mentions `#REQUIRED` et `#IMPLIED`.

```
< !ATTLIST scene moment (jour|nuit) #REQUIRED
                decor (intérieur|extérieur) #REQUIRED
                comediens CDATA #IMPLIED >
```

Procéder de la sorte garantira la présence d'informations jugées indispensables. Dans cet exemple, il sera possible de faire des recherches directement sur la valeur des attributs.

Une nomenclature stricte permet d'accéder à n'importe quelle base à travers une identification relative. Il est également possible d'affiner l'identification des nœuds à travers l'exploitation de l'arborescence structurant les balises grâce à une étude de leur généalogie.

2.3. XPath, un fil d'Ariane

Une personne peut se définir par un certain nombre de traits qui lui sont propres. Les attributs de balises servent en ce sens à caractériser un élément. Mais il est fréquent, en particulier dans les petites structures sociales, qu'une personne soit caractérisée par le lien qu'elle possède avec un ou des ascendants. Il en va de même dans une arborescence XML.

```
<racine>

  <titre>Le Cid</titre>

  <acte>
    <scene>Elvire et Chimène</scene>
    <scene>Doña Urraque amoureuse de Rodrigue</scene>
    <scene>Don Diègue au poste de gouverneur</scene>
    <scene>Monologue de Don Diègue</scene>
    <scene>Volonté de tuer Don Gomès</scene>
    <scene>Monologue de Rodrigue</scene>
  </acte>

</racine>
```

Dans cet exemple, il est possible de retrouver n'importe quelle balise via le chemin qui mène à elle depuis la racine. Dans cette généalogie, la donnée textuelle est représentée par `text()`. Ainsi, « Le Cid » correspond à `racine/titre/text()`, « Elvire et Chimène » à `racine/acte/scene/text()`. S'il n'existe pas d'ambiguïté pour « Le Cid », car chaque élément du chemin est unique dans l'arborescence, le chemin menant à « Elvire et Chimène » peut sembler plus équivoque, car il existe plusieurs balises `scene` dans `acte`. Si ce chemin renvoie à « Elvire et Chimène » et pas à « Doña Urraque amoureuse de Rodrigue », c'est parce que dans la descendance de `acte`, le `scene` contenant « Elvire et Chimène » est le premier de cette génération et ce sera lui qui sera considéré comme le fils par défaut sans autre mention. Il sera donc possible d'accéder à la donnée du second `scene` à condition de le caractériser par son rang dans sa génération. Le chemin pour atteindre la donnée « Doña Urraque amoureuse de Rodrigue » sera donc `racine/acte/scene[2]/text()`.

Il est également possible d'accéder à cette donnée en mêlant un chemin absolu et un chemin relatif. Il est en effet possible d'accéder à des balises à travers d'autres de la même génération. Il est possible d'accéder à cette même donnée en se servant de la première balise `scene` comme d'un relais après une exploration verticale pour ensuite procéder à une exploration horizontale grâce à des références relatives dans la génération, par exemple par le biais de `following-sibling::*` pour la balise suivante et de `previous-sibling::*` pour la précédente :

`racine/acte/scene/text()` « Elvire et Chimène »

`racine/acte/scene[3]/text()`
« Don Diègue au poste de gouverneur »

`racine/acte/scene[3]/following-sibling::*/text()`
« Monologue de Don Diègue »

`racine/acte/scene[3]/previous-sibling::*/text()`
« Doña Urraque amoureuse de Rodrigue »

Il est également possible de se servir des attributs pour injecter de la détermination relative dans un chemin absolu. Si l'on considère cet extrait du fichier XML présenté en 1.3.1.2., il est possible d'accéder à une donnée précise en s'appuyant sur les attributs de balise :

```
<racine>

  <titre>Le Cid</titre>

  <acte id=”1”>
    <scene id=”1”>Elvire et Chimène</scene>
    <scene id=”2”>Doña Urraque amoureuse de Rodrigue</scene>
    <scene id=”3”>Don Dièguе au poste de gouverneur</scene>
    <scene id=”4”>Monologue de Don Dièguе</scene>
    <scene id=”5”>Volonté de tuer Don Gomès</scene>
    <scene id=”6”>Monologue de Rodrigue</scene>
  </acte>

</racine>
```

L’accès au titre de la cinquième scène se fera via le chemin suivant :

```
racine/acte/scene/@id=”5”/text()
```

L’attribut apparaît ici comme une balise d’un type bien particulier mais balise malgré tout.

La conjonction de ces moyens permet d’accéder à la donnée de manière rapide en la ciblant directement. XPath permet également d’améliorer les performances d’accès à la donnée en se servant de ces moyens d’identification pour éliminer le bruit et spécifier un chemin en réduisant le nombre cartésien de balises à l’aide de subordonnées, exprimées par les crochets droits. L’apport des attributs de balise prend alors tout son sens. Dans le fichier XML reprenant la structure du *Cid*, il est très aisé d’accéder directement au titre de la scène simplement en caractérisant les nœuds quand nécessaire. En langue naturelle, « La mort de Don Gomès » est la scène 7 du deuxième acte. Le chemin menant à cette donnée sera donc :

```
racine [./titre=”Le Cid”] / acte [./@id=”2”] / scene
[./@id=”7”] / text()
```

Chaque partie entre crochet est une subordonnée qui sert à déterminer le nœud en question. Dans des structures complexes où l’accès à la donnée peut prendre plusieurs minutes voire dizaines de minutes, réduire le nombre de chemins possibles a un impact certain sur les performances de gestion de la base de données.

L'ensemble de ces technologies et fonctionnalités satellites du XML sont autant de briques qui vont par la suite permettre de construire un corpus et de l'interroger en tirant partie de ces couches d'abstraction. En convertissant la structure d'un texte en arbre XML, ces briques vont être à la base de requête qui opéreront un traitement informatique de données linguistiques et inversement grâce au langage de requêtage XQuery.

2.4. XQuery, à la recherche de la donnée

XQuery est un langage de requêtage, au même titre que SQL, appartenant au paradigme fonctionnel. Un langage fonctionnel n'altère pas la donnée, elle est le facteur d'une opération de transformation afin de procéder à son exploitation. Un des emplois les plus communs de XQuery est la récupération de données dans une base XML. La version actuelle est la 3.0[8]. XQuery a été développé et pensé pour le XML et ses fonctions natives en font un outil particulièrement adapté pour le traitement de la langue. En tant que langage de requêtage, XQuery accède aux données stockées dans une base mais permet de sculpter ses ressources de manière *ad hoc*.

2.4.1. La boucle au cœur de la requête

À l'inverse de Python, PERL ou C qui sont des langages procéduraux, XQuery s'inscrit dans le paradigme fonctionnel. XQuery est un langage de requêtage. Il permet d'interroger une base de données comme SQL le fait avec des tables. Une des différences entre un fichier XML et une table SQL est la profondeur de la structure. Afin d'accéder aux

[8] Plus d'informations sur les spécificités de XQuery sur le site du W3C, http://www.w3.org/TR/2014/REC-xquery-30-20140408/#id-introduction.

données, les requêtes se basent sur l'utilisation de boucles. On dénombre cinq instructions regroupées sous l'acronyme FLOWR :

- `for`
- `let`
- `order by`
- `where`
- `return`

Les deux instructions les plus importantes sont `for` et `return`. `for` lance la boucle qui parcourt le fichier XML où la variable de boucle va prendre successivement plusieurs états et `return` renvoie le résultat de la requête. L'arbre XML suivant est un extrait du dictionnaire électronique Morphalou. Il est possible avec une simple requête d'obtenir tous les lemmes du dictionnaire :

```
<lexicon>
  <lexicalEntry>
    <lemmatizedForm>
      <orthography>casserole</orthography>
      <grammaticalCategory>commonNoun</grammaticalCategory>
      <grammaticalGender>feminine</grammaticalGender>
    </lemmatizedForm>
    <inflectedForm>
      <orthography>casserole</orthography>
      <grammaticalNumber>singular</grammaticalNumber>
    </inflectedForm>
    <inflectedForm>
      <orthography>casseroles</orthography>
      <grammaticalNumber>plural</grammaticalNumber>
    </inflectedForm>
  </lexicalEntry>
  <lexicalEntry>
    <lemmatizedForm>
      <orthography>alligator</orthography>
      <grammaticalCategory>commonNoun</grammaticalCategory>
      <grammaticalGender>masculine</grammaticalGender>
    </lemmatizedForm>
    <inflectedForm>
      <orthography>alligator</orthography>
      <grammaticalNumber>singular</grammaticalNumber>
    </inflectedForm>
    <inflectedForm>
      <orthography>alligators</orthography>
      <grammaticalNumber>plural</grammaticalNumber>
```

```
    </inflectedForm>
  </lexicalEntry>
  <lexicalEntry>
    <lemmatizedForm>
      <orthography>petit</orthography>
      <grammaticalCategory>adjective</grammaticalCategory>
    </lemmatizedForm>
    <inflectedForm>
      <orthography>petit</orthography>
      <grammaticalNumber>singular</grammaticalNumber>
      <grammaticalGender>masculine</grammaticalGender>
    </inflectedForm>
    <inflectedForm>
      <orthography>petits</orthography>
      <grammaticalNumber>plural</grammaticalNumber>
      <grammaticalGender>masculine</grammaticalGender>
    </inflectedForm>
    <inflectedForm>
      <orthography>petite</orthography>
      <grammaticalNumber>singular</grammaticalNumber>
      <grammaticalGender>feminine</grammaticalGender>
    </inflectedForm>
    <inflectedForm>
      <orthography>petites</orthography>
      <grammaticalNumber>plural</grammaticalNumber>
      <grammaticalGender>feminine</grammaticalGender>
    </inflectedForm>
  </lexicalEntry>
</lexicon>


for $x in lexicon/lexicalEntry/lemmatizedForm/orthography
return $x
```

Lors du parcours de la boucle, `$x` va successivement prendre la valeur de tous les nœuds qui correspondent à ce chemin XPath. Cette liste sera l'élément renvoyé par `return`. Ainsi, le résultat de cette requête sera :

```
<orthography>casserole</orthography>
<orthography>alligator</orthography>
<orthography>petit</orthography>
```

2.4.2. Construire la requête idoine

L'un des atouts majeurs de XQuery est de pouvoir sculpter soi-même le résultat que l'on veut obtenir. Il est tout à fait possible de reconstruire un arbre XML à partir des données collectées. La fonction `data()` contribue à cette possibilité en accédant directement à la

donnée contenue dans les balises. Les instructions entre accolades permettent de mêler ce qui est de la donnée et ce qui doit être interprété comme du code XQuery.

```
<lexique>
{ for $x in lexicon/lexicalEntry/lemmatizedForm/orthography
return <lemme>{data($x)}</lemme> }
</lexique>
```

Cette requête nous permettra d'obtenir notre propre ressource linguistique, un dictionnaire composé de lemmes uniquement :

```
<lexique>
  <lemme>casserole</lemme>
  <lemme>alligator</lemme>
  <lemme>petit</lemme>
</lexique>
```

La fonction `string-join()` crée une chaîne de caractères à partir du contenu d'une balise. `text()` renvoie la donnée, mais seulement les chaînes de caractères qui sont contenues dans la balise concernée. Si elle contient une balise elle-même, son contenu sera ignoré. `data()` permet d'accéder directement à tous les contenus jusqu'à la plus grande profondeur possible. `string-join()` permet de reconstruire une chaîne de caractère à partir d'un arbre XML :

```
<descr sens="temporalité">attendre longtemps</descr>
  <exp registre="courant">
    <v>attendre</v>
    <adv>
      <det>107</det>
      <n>ans</n>
    </adv>
  </exp>
</descr>
```

La requête suivante permet de récupérer l'expression figée de ce dictionnaire :

```
for $x in descr/exp
return<exemple>{string-join($x//text()," ")}</exemple>
```

Le résultat renvoyé est:

```
<exemple>attendre 107 ans</exemple>
```

L'exploitation de XPath permet de trier les entrées pour obtenir un résultat particulier. Ainsi, si l'on ne souhaite obtenir que des lemmes appartenant à la catégorie des adjectifs, il suffit d'adapter la requête soit en ajoutant une clause *where* soit en se servant des crochets droits. Voici deux requêtes synonymes pour obtenir ce résultat :

```
<lexique>
{ for $x in lexicon/lexicalEntry/lemmatizedForm
where $x/grammaticalCategory='adjective'
return <lemme>{data($x/orthography)}</lemme> }
</lexique>
```

et

```
<lexique>
{ for $x in
lexicon/lexicalEntry/lemmatizedForm[./grammaticalCategory='ad
jective']
return <lemme>{data($x/orthography)}</lemme> }
</lexique>
```

Ces deux requêtes produiront le même résultat : un lexique de formes lemmatisées d'adjectifs. De nombreuses fonctions permettent de créer des requêtes spécifiques à la recherche. L'utilisation des expressions rationnelles est possible avec la fonction `matches()`. Elle autorise de créer un motif ou des motifs qui pourront être exploités dans la requête. Par exemple `matches(.,'poi.?res?')` reconnaîtra potentiellement `poire, poires, poivre, poivres`.

Il est possible en XQuery de déclarer des variables comme dans les langages procéduraux. Couplée aux expressions rationnelles, la déclaration de variables permet de démultiplier les candidats avec l'instruction `let`. La nature de la variable peut être simple, comme une chaîne de caractères, mais il est également possible de déclarer une liste :

```
let $liste := ("commonNoun","verb")
```

En initialisant une boucle qui parcourra `$liste` pour prendre successivement chacune de ses valeurs, on introduit une couche de potentialité à la requête.

```
let $liste := ("commonNoun","verb")
return
<lexique>
{
for $x in
lexicon/lexicalEntry/lemmatizedForm[./grammaticalCategory=$li
ste]
return <lemme>{data($x/orthography)}</lemme> }
```

```
</lexique>
```

On obtient ainsi un lexique composé des lemmes de tous les noms communs et de tous les verbes du dictionnaire. Il est possible d'obtenir un seul résultat pour plusieurs requêtes.

La maîtrise de la structure est une clé de l'exploitation d'un document XML. Celle de la donnée en est une autre. Une chaîne de caractères est un flux de données. Afin de pouvoir opérer un traitement linguistique de ce flux, il est indispensable de pouvoir d'en isoler ses unités constitutives. La définition du mot pose problème ; l'informatique l'a contourné via le token. Un token se définit comme un ensemble de caractères séparés par un caractère spécifique, l'espace est généralement celui retenu pour se rapprocher d'un découpage en mots relativement performant. Ce processus s'appelle la tokenisation. Il est également possible d'aller à un niveau supérieur et de procéder à une tokenisation par phrases en prenant comme caractère séparateur toute marque de ponctuation forte, comme le point, le point d'exclamation, d'interrogation, les points de suspension, etc. Une fois que nous disposons d'un texte segmenté, il devient possible de mettre au point rapidement un étiqueteur morphosyntaxique en projetant un dictionnaire en XML pour créer des correspondances avec les tokens. La segmentation en phrases permet aussi de reconstruire automatiquement la structure XML d'un texte en identifiant automatiquement les phrases, les paragraphes et les chapitres.

XQuery possède une fonction qui permet de segmenter ce flux : `tokenize()`. Il est ainsi très simple de récupérer dans une variable tout un texte tokenisé ou segmenté en phrases depuis n'importe quelle balise d'un fichier XML :

```
let $corpus := tokenize(/corpus, "\.|\.\.\.|\?|!")
```

Une fois le corpus tokenisé, il devient possible de projeter des ressources linguistiques, un dictionnaire comme Morphalou par exemple, et d'opérer un étiquetage morphosyntaxique. La première étape est de tokeniser le texte sur l'espace et les signes de ponctuation afin d'isoler des unités :

```
let $corpus := tokenize(/corpus," |\.|\.\.\.|\?|!\,|;|'")
```

La fonction `distinct-values()` permet de créer un index en dressant une liste de tokens sans doublons :

```
let $index := distinct-values($corpus)
```

La requête porte sur chaque mot de l'index (`$x`) *et* sur chaque entrée orthographique où la forme fléchie (`$y`) correspond à ce mot (`$x`). L'emploi de la virgule permet de croiser des bases de données. Il ne s'agit pas de lancer deux boucles successivement, mais une seule qui passera la requête aux deux bases de données et qui renverra l'ensemble partagé par les deux :

```
for $x in $index,
$y in db:open('morphalou')/lexicon/lexicalEntry
[./formSet/inflectedForm/orthography=$x]
```

L'instruction `order by` permet de procéder à un tri selon des critères définis. Ici, le tri se fera alphabétiquement sur `$y`, soit les entrées orthographiques du dictionnaire correspondant à un token présent dans le corpus.

```
order by $y
```

Le résultat renvoyé est l'apposition d'une balise `<analyse>` avec pour premier attribut `lemme` dont la valeur est le lemme de Morphalou, pour deuxième attribut `token` le mot-forme issu du texte et pour troisième attribut `pos` qui contient la catégorie grammaticale extraite du dictionnaire :

```
return


element analyse {attribute lemme
{$y/formSet/lemmatizedForm/orthography} (:le lemme de
Morphalou:), attribute token {$x}, attribute pos
{$y/formSet/lemmatizedForm/grammaticalCategory}}
```

XML est conçu pour l'exploitation de données textuelles. Les opérations précédentes peuvent trouver un équivalent dans des langages du paradigme impératif comme Python ou PERL. Ce dernier est particulièrement adapté à l'emploi d'expressions rationnelles. Ces langages peuvent très bien produire du XML : le concordancier de Corpindex, qui est programmé en Python 3, propose nativement un export des concordances dans un arbre XML créé à la volée. L'atout majeur de l'orientation de XQuery vers le texte est la mise à la disposition de nombreux modules plein-texte qui permettent un traitement linguistique d'un flux de données numérique.

2.4.3. Des outils morphologiques et morphosyntaxiques : les modules plein-texte

XQuery possède des avantages certains dans le traitement de données textuelles : les tokens ne sont pas simplement considérés comme des chaines de caractères mais comme des unités lexicales. Un ouvrage comme *Perl pour les Linguistes*[9] aide au traitement de la langue écrite au moyen d'un langage de programmation. PERL est un outil permettant de nombreuses transformations du texte, mais reste à un niveau de surface, en considérant des tokens. Les modules plein-texte mettent à disposition de l'humaniste numérique tout un panel d'outils pour un traitement plus naturel de la langue. XQuery est connu pour être verbeux, cela peut être vu comme un défaut ou une qualité. L'avantage que cette verbosité offre est de proposer des ponts syntaxiques et lexicaux entre langage de programmation et langue naturelle, en l'occurrence l'anglais.

La recherche d'un mot peut se faire au moyen d'expressions rationnelles, comme vu précédemment avec la fonction `matches()`. Une expression rationnelle est un motif de caractères proposant des options selon une nomenclature déterminée[10]. XQuery propose de procéder à une recherche non pas à travers le jeu de caractères qui le compose mais en le considérant comme une unité lexicale. Il est tout à fait possible de chercher le mot « poire » avec `matches()` :

```
for $x in corpus/phrase[./ matches(., 'chat')]
return $x
```

Il sera plus efficace de lancer une requête en employant directement le module plein-texte dédié :

```
for $x in corpus/phrase[./ contains text('chat') any word]
return $x
```

[9] Ludovic TANGUY, Nabil HATHOUT, et al, *Perl pour les linguistes*. Paris : Hermes Science Publications, 2007.
[10] *Ibid*.

À partir de ce module, il est possible de spécifier l'empan dans lequel le résultat doit être cherché en précisant `any word`, `all words`, `any sentence`, `all sentence`, `any paragraph`, `all paragraph`.

L'exploitation linguistique du texte se fait également en tirant partie des propriétés morphologiques de l'unité lexicale. Trouver « chat » au moyen d'une expression rationnelle est relativement simple : `/^chat$/`. Il est aussi aisé de procéder à celle de « chat » au pluriel et au singulier : `/^chats?$/`. Si envisager la morphologie d'un mot semble peu contraignant quand il s'agit de noms à pluriel régulier, l'expression deviendra très rapidement difficile à écrire et encore plus vite illisible lorsqu'il s'agira de verbes. XQuery propose de faire des recherches en s'appuyant sur les racines, ou *stem* en anglais.

```
for $x in corpus/phrase[./ contains text('chat') any word
using stemming]
return $x
```

La langue est du document est spécifiée au moment de l'indexation mais il est également possible de la préciser a posteriori.

```
for $x in corpus/phrase[./ contains text('chat') any word
using stemming using language 'fr']
return $x
```

Il est alors possible de combiner plusieurs paramètres en multipliant le nombre de mots cherché avec `ftand` et `ftor`, de créer une liste de mots à éviter avec `ftnot` dans un empan avec `distance` :

```
for $x in corpus/phrase[./ contains text 'chat' ftand 'chien'
ftor 'loup' all words same sentence using stemming using
language 'fr' distance at most 3 words]
return $x
```

Deux modules présentant des similarités permettent d'agir directement sur la donnée textuelle en combinaison avec les éléments de détermination précédemment abordés. `ft:mark` appose des balises automatiquement sur les éléments recherchés. Les balises `<mark>` sont génériques et permettent un étiquetage sommaire :

```
<racine>
{
let $test := /corpus
return ft:mark($test/p[./text() contains text
{'homme','sédentaire'} all same sentence])
}
</racine>
```

La requête ci-dessus lancée sur *Le tour du monde en 80 jours* dont la racine est `corpus` et les paragraphes sont entre balises `<p>` renverra un nouveau fichier XML où les mots « homme » et « sédentaire » apparaissent dans la même phrase. Il appose par défaut la balise `<mark>` mais offre l'option de créer des balises personnalisées sur les mots du texte :

<racine>
- Jean, n'en déplaise à monsieur, répondit le nouveau venu, Jean Passepartout, un surnom qui m'est resté, et que justifiait mon aptitude naturelle à me tirer d'affaire. Je crois être un honnête garçon, monsieur, mais, pour être franc, j'ai fait plusieurs métiers. J'ai été chanteur ambulant, écuyer dans un cirque, faisant de la voltige comme Léotard, et dansant sur la corde comme Blondin ; puis je suis devenu professeur de gymnastique, afin de rendre mes talents plus utiles, et, en dernier lieu, j'étais sergent de pompiers, à Paris. J'ai même dans mon dossier des incendies remarquables. Mais voilà cinq ans que j'ai quitté la France et que, voulant goûter de la vie de famille, je suis valet de chambre en Angleterre. Or, me trouvant sans place et ayant appris que M. Phileas Fogg était l'<mark>homme</mark> le plus exact et le plus <mark>sédentaire</mark> du Royaume-Uni, je me suis présenté chez monsieur avec l'espérance d'y vivre tranquille et d'oublier jusqu'à ce nom de Passepartout...
</racine>

La même requête peut faire un marquage des unités mais également sélectionner un empan en ajoutant `distance` :

```
<racine>
{
let $test := /corpus
return ft:mark($test/p[./text() contains text
{'homme','sédentaire'} all same sentence distance at least 3
words])
}
</racine>
```

La balise est donc apposée sur tous les tokens présents entre « homme » et « sédentaire » si les deux sont séparés par au moins trois mots et s'ils sont dans la même phrase :

<racine>
- Jean, n'en déplaise à monsieur, répondit le nouveau venu, Jean Passepartout, un surnom qui m'est resté, et que justifiait mon aptitude naturelle à me tirer d'affaire. Je crois être un honnête garçon, monsieur, mais, pour être franc, j'ai fait plusieurs métiers. J'ai été chanteur ambulant, écuyer dans un cirque, faisant de la voltige comme Léotard, et dansant sur la corde comme Blondin ; puis je suis devenu professeur de gymnastique, afin de rendre mes talents plus utiles, et, en dernier lieu, j'étais sergent de pompiers, à Paris. J'ai même dans mon dossier des incendies remarquables. Mais voilà cinq ans que j'ai quitté la France et que, voulant goûter de la vie de famille, je suis valet de chambre en Angleterre. Or, me trouvant sans place et ayant appris que M. Phileas Fogg était l'<mark>homme</mark> <mark>le</mark> <mark>plus</mark> <mark>exact</mark> <mark>et</mark> <mark>le</mark> <mark>plus</mark> <mark>sédentaire</mark> du Royaume-Uni, je me suis présenté chez monsieur avec l'espérance d'y vivre tranquille et d'oublier jusqu'à ce nom de Passepartout...
</racine>

`ft:extract` a un comportement similaire à `ft:mark`, la différence réside dans le résultat, qui sera un extrait et non le texte entier. Le dernier argument est la taille du contexte à inclure dans l'extraction :

```
<racine>
{
let $corpus := /corpus
return ft:extract($corpus/p[./text() contains text
{'Passepartout','Phileas'} all same sentence distance at most
3 words],'extract',100)
}
</racine>
```

Si la requête n'avait comporté qu'un seul mot à extraire, il aurait été le seul balisé avec `<extract>`. L'extraction sur plusieurs mots produit un résultat similaire à un `ft:mark` sur plusieurs mots et les traite comme le début et la fin d'un empan :

<racine>

```
<p>...ANS LEQUEL <extract>PHILEAS</extract>
<extract>FOGG</extract> <extract>ET</extract>
<extract>PASSEPARTOUT</extract> S'ACCEPTENT
RÉCIPROQUEMENT L'UN COMME MAÎTRE, L'AUTRE COMM...</p>

<p>DANS LEQUEL <extract>PHILEAS</extract>
<extract>FOGG</extract> <extract>STUPEFIE</extract>
<extract>PASSEPARTOUT</extract>, SON
DOMESTIQUE</p>

<p>... salle de la gare. Là, <extract>Phileas</extract>
<extract>Fogg</extract> <extract>donna</extract>
<extract>à</extract> <extract>Passepartout</extract>
l'ordre de prendre deux billets de premiè...</p>

<p>DANS LEQUEL <extract>PHILEAS</extract>
<extract>FOGG</extract>,
<extract>PASSEPARTOUT</extract>, FIX, CHACUN DE SON
CÔTÉ, VA A SES AFFAIRES</p>

</racine>
```

La requête extrait des paragraphes les segments d'une longueur de cent caractères qui débutent par « Phileas » et se terminent par « Passepartout » dans la même phrase et dans un espace de trois mots.

Afin d'étendre les candidats d'une recherche, il existe une implémentation native de l'algorithme de Levenshtein : `fuzzy`. À partir d'un mot donné, il est possible de considérer en plus de celui-ci toutes les transformations mettant en jeu un caractère : une suppression, une insertion ou un remplacement. Ainsi, la requête suivante permet de démultiplier les candidats :

```
<racine>
{
let $corpus := /corpus
return ft:extract($corpus/p[./text() contains text {"sacré"}
using fuzzy],'extract',100)
}
</racine>
```

Le résultat obtenu illustre les mots se rapprochant de « sacré » à une distance de Levenshtein de un :

```
<racine>
```

. . . ceci : autrefois, dans l'Inde, les chats étaient considérés comme des animaux <extract>sacrés</extract>. C'était . . .

...doue, et cela, en le nourrissant pendant trois mois de <extract>sucre</extract> et de beurre. Ce traitement peut paraî . . .

... ses sauts de carpe, et, de temps en temps, il tirait de son sac un morceau de <extract>sucre</extract>, que l'intellig . . .

...ar des torches fuligineuses, veillaient aux portes et se promenaient, le <extract>sabre</extract> nu. On pouvait suppos . . .

...le est bâtie au confluent de deux fleuves <extract>sacrés</extract>, le Gange et la Jumna, dont les eaux attirent les . . .

...s, nagent comme dans les lacs <extract>sacrés</extract> de l'Himalaya les reflets les plus purs de la lumière célest...

...se baigner dans les eaux du fleuve <extract>sacré</extract>, et aussi, malgré la saison avancée et la température d . . .

...ant de la loi, et, pour tout Anglais, la loi est <extract>sacrée</extract>. Passepartout, avec ses habitudes français . . .

...de Mr. Fogg, ses grands yeux " limpides comme les lacs <extract>sacrés</extract> de l'Himalaya " ! Mais l'intraitable . . .

...bas, avec une toupie tournante sur la plante du pied gauche, et un <extract>sabre</extract> en équilibre sur la plante . . .

</racine>

Le repérage et le marquage d'unités lexicales est utile, mais il peut être pertinent d'en extraire une information numérique. XQuery permet d'évaluer le poids d'un mot dans une phrase réduit à un nombre entre 0 et 1 :

```
<racine>{
let $test := /corpus

for $x score $score in $test/p[./text() contains text
'whist']
order by $score descending
```

```
return <res score="{$score}">{data($x)}</res>
}
</racine>
```

Le résultat correspond au classement des paragraphes contenant le texte « whist » selon le poids de celui-ci dans le paragraphe :

```
<racine>
  <res score="0.10845263981822655">- A bord des paquebots, reprit l'inspecteur, vous aviez l'habitude de faire votre whist ?</res>

  <res score="0.09117445910813965">- Certainement, monsieur, répondit vivement la jeune femme, je connais le whist. Cela fait partie de l'éducation anglaise.</res>

  <res score="0.0865775114495933">- Tout compris, répondit Phileas Fogg en continuant de jouer, car, cette fois, la discussion ne respectait plus le whist .</res>

  <res score="0.07520368258519186">Sept heures sonnaient alors. On offrit à Mr. Fogg de suspendre le whist afin qu'il pût faire ses préparatifs de départ .</res>

  <res score="0.07520368258519186">Mr. Fogg tira de sa poche les vingt guinées qu'il venait de gagner au whist, et, les présentant à la mendiante :</res>

  <res score="0.06399565127886572">" Bon, fit-il, Mrs. Aouda est encore couchée à cette heure. Quant à Mr. Fogg, il aura trouvé quelque joueur de whist, et suivant son habitude... "</res>

  <res score="0.05955789604015849">Les voyageurs étaient réintégrés dans leur wagon. Passepartout reprit sa place, sans rien dire de ce qui s'était passé. Les joueurs étaient tout entiers à leur whist.</res>

  <res score="0.04930212912827189">Après un déjeuner assez confortable, servi dans le wagon même, Mr. Fogg et ses partenaires venaient de reprendre leur interminable whist, quand de violents coups de sifflet se firent entendre. Le train s'arrêta.</res>

  <res score="0.01955785797433551">Une demi-heure plus tard, divers membres du Reform-Club faisaient leur entrée et s'approchaient de la cheminée, où brûlait un feu de houille. C'étaient les partenaires habituels de Mr. Phileas Fogg, comme lui enragés joueurs de whist : l'ingénieur Andrew Stuart, les banquiers John Sullivan et Samuel Fallentin, le brasseur Thomas Flanagan, Gauthier Ralph, un des administrateurs de la
```

Banque d'Angleterre, - personnages riches et considérés, même dans ce club qui compte parmi ses membres les sommités de l'industrie et de la finance.</res>

<res score="0.01109588654349911">Ce qui était certain toutefois, c'est que, depuis de longues années, Phileas Fogg n'avait pas quitté Londres. Ceux qui avaient l'honneur de le connaître un peu plus que les autres attestaient que - si ce n'est sur ce chemin direct qu'il parcourait chaque jour pour venir de sa maison au club - personne ne pouvait prétendre l'avoir jamais vu ailleurs. Son seul passe-temps était de lire les journaux et de jouer au whist. A ce jeu du silence, si bien approprié à sa nature, il gagnait souvent, mais ses gains n'entraient jamais dans sa bourse et figuraient pour une somme importante à son budget de charité. D'ailleurs, il faut le remarquer, Mr. Fogg jouait évidemment pour jouer, non pour gagner. Le jeu était pour lui un combat, une lutte contre une difficulté, mais une lutte sans mouvement, sans déplacement, sans fatigue, et cela allait à son caractère.</res>

</racine>

Le résultat illustre à travers le score la part qu'a le mot dans le segment. Il apparaît que plus le mot représente une portion importante du paragraphe, plus le score sera élevé. Inversement, plus le mot représente une part marginale du paragraphe, plus le score sera faible.

L'intégration des expressions rationnelles est également présente dans les modules plein-texte à travers *wildcards*. Il est en effet possible d'utiliser dans les unités à chercher des expressions régulières qui seront alors interprétées comme telles :

```
let $regex := "imp.*able"

return
ft:mark(corpus/p[./text() contains text {$regex} all using
wildcards],'adj')
```

Ce fut même ce qui arriva dans cette occasion. Vers trois heures du soir, un troupeau de dix à douze mille têtes barra le rail-road. La machine, après avoir modéré sa vitesse, essaya d'engager son éperon dans le flanc de l'immense colonne, mais elle dut s'arrêter devant l'<adj>impénétrable</adj> masse.

- Je ferai tout pour le ramener vivant en Europe ! " répondit simplement Fix, d'un ton qui marquait une <adj>implacable</adj> volonté.

Quant à Passepartout, la face rouge comme le disque solaire quand il se couche dans les brumes, il humait cet air piquant. Avec le fond d'<adj>imperturbable</adj> confiance qu'il possédait, il s'était repris à espérer. Au lieu d'arriver le matin à New York, on y arriverait le soir, mais il y avait encore quelques chances pour que ce fût avant le départ du paquebot de Liverpool .

Les modules plein-texte sont un apport certain à l'exploitation informatique des Humanités numériques. Tous ces modules apportent chacun leurs spécificités mais peuvent également se combiner pour donner une programmation très proche de la langue naturelle. Le dernier module plein-texte que nous allons présenter est celui sur lequel s'appuiera les applications qui seront envisagées par la suite : le thésaurus.

2.4.4. Le traitement sémantique des unités lexicales : le thésaurus

Le thésaurus est une structure proposée par XQuery pour établir des liens sémantiques entre unités lexicales à travers un jeu de relations. Le thésaurus doit se conformer à un schéma XML[11]. Pour créer un thésaurus exploitant au mieux les fonctionnalités proposées par XQuery, la racine est obligatoirement <thesaurus> dans laquelle vont s'insérer des balises <entry>. Chaque <entry> est composé d'un terme pivot autour duquel vont se construire les relations sémantiques et hiérarchiques :

```
<thesaurus>
   <entry>
      <term>céréale</term>
      <synonym>
         <term>blé</term>
         <relationship>NT</relationship>
      </synonym>
      <synonym>
         <term>riz</term>
         <relationship>NT</relationship>
      </synonym>
      <synonym>
         <term>orge</term>
         <relationship>NT</relationship>
      </synonym>
```

[11] Voir Annexe 3.

```
        <synonym>
           <term>froment</term>
           <relationship>NT</relationship>
        </synonym>
        <synonym>
           <term>sorgho</term>
           <relationship>NT</relationship>
        </synonym>
     </entry>
</thesaurus>
```

Les relations suivent une nomenclature déterminée par le schéma XML du W3C. `NT` correspond à un hyponyme, `BT` à un hyperonyme et `TT` à une hyperclasse.

L'utilisation du module plein-texte `thesaurus` se fait au sein même de la requête. Le thésaurus est un fichier XML qui va être appelé en spécifiant son adresse locale :

```
<racine>
{
let $test := /corpus

return
ft:extract($test/p[./text() contains text "céréale" using
thesaurus at "C:\XML\thesaurus.xml" relationship 'NT' at most
2 levels ordered],'cereale',100)
}
</racine>
```

La requête concerne donc la recherche du mot « céréale » et permet de le substituer par n'importe quel hyponyme jusqu'à deux niveaux de profondeur dans le thésaurus.

```
<racine>

<p>...de quelques poignées de <cereale>riz</cereale>, on la
repousserait, elle serait considérée comme une créature immon...</p>

<p>...issait le paysage varié du Béhar, puis des montagnes couvertes de
verdure, les champs d'<cereale>orge</cereale>, de m...</p>

<p>... pleine tasse l'eau chaude odorante, avec le " saki ", liqueur tirée du
<cereale>riz</cereale> en fermentation, et ces...</p>

<p>...rence, et là, d'un reste de volaille et de quelques poignées de
<cereale>riz</cereale>, il déjeuna en homme pour qui...</p>
```

```
<p>...aux de la terre : des ranchos et des corrals pour les animaux
domestiques, des champs de <cereale>blé</cereale>, de
ma...</p>

</racine>
```

Le thésaurus constitue un point d'entrée dans les réseaux lexicaux avec une mise en correspondance à travers des relations hiérarchiques et sémantiques d'unités lexicales. Ce point constituera l'une des perspectives de ce mémoire.

2.5. XML et le web : un corpus à dimension mondiale

XML est lié au web de par la nature des données qu'ils sont amenés à traiter. Si internet permet d'avoir accès à tous types de données, le web est résolument orienté texte. En effet, un site web actuel ne se conçoit plus comme dans les années 90. Le web est aujourd'hui multimédia, les sons côtoient les images et la mise en page est dynamique. Mais au cœur est le texte. Une page web n'est composée que de texte[12]. Le développement du web a été participatif, ce qui a conduit à une introduction de flexibilité pour pouvoir le démocratiser. XML et son formalisme sont vus à l'inverse comme des facilités permises par l'interprétation du HTML par les navigateurs web. En 2000, d'une volonté d'apporter plus de rigueur à l'écriture du web est né XHTML.

2.5.1. XHTML

Le XHTML se distingue du HTML 4 qui était développé à la même époque par une plus grande volonté de rigueur. La syntaxe d'une page en XHTML est très proche d'une page

[12] Voir Annexe 1.

en HTML où le formalisme XML serait scrupuleusement respecté et où la donnée serait rigoureusement ordonnée.

Voici un exemple de page html didactisée[13] illustrant les dérives tolérées en HTML :

```
<html>
  <head>
    <title>This is bad HTML</title>
  <body>
    <h1>Bad HTML
    <br>
    <p>This is a paragraph
  </body>
```

En XHTML, le principe des balises est un impératif. Ici, La balise `<html>` n'est pas fermée, il n'existe donc pas de racine, tout comme `<head>`, `<h1>` et `<p>`. Voici la même page mais avec des balises respectant le formalisme XML :

```
<html>
  <head>
    <title>This is bad HTML</title>
  </head>
  <body>
    <h1>Bad HTML</h1>
  <br/>
    <p>This is a paragraph</p>
  </body>
</html>
```

XHTML ajoute de la contrainte rédactionnelle par le strict respect des règles suivantes :

- Il ne peut y avoir qu'une seule racine
- les balises ne peuvent plus se chevaucher
- le nom des balises doit être en minuscule
- la valeur des attributs est entre guillemets
- un DOCTYPE doit être déclaré (DTD)

[13] Extrait de http://www.w3schools.com/html/html_xhtml.asp.

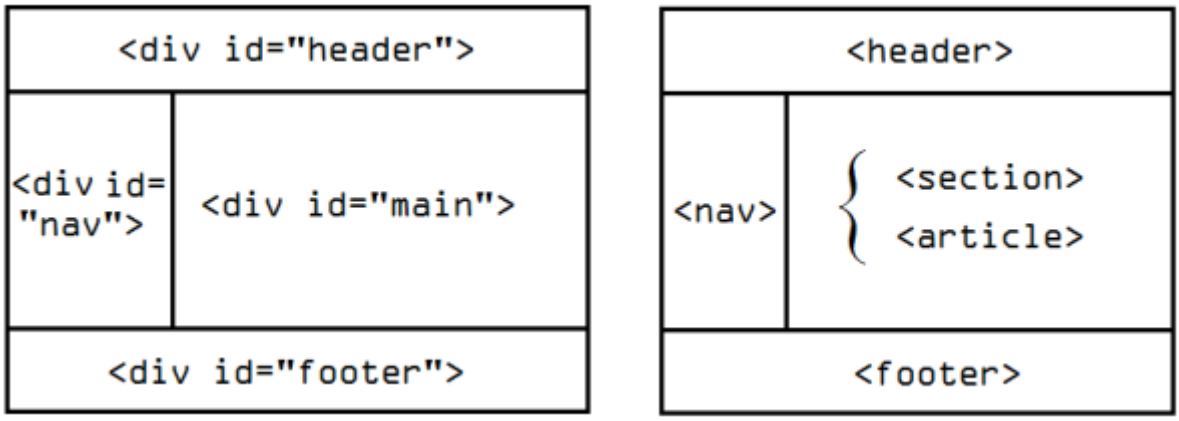
Cette tentative d'instaurer de la rigueur sur le web n'a que peu été suivie d'effets. XHTML n'a pas su s'imposer comme standard, les rédacteurs lui ont préféré HTML, qui depuis a évolué vers sa cinquième version. Le HTML 5 n'a pas poursuivi les objectifs du XHTML mais permet le développement du web sémantique.

2.5.2. *Big Data* et web sémantique

Le web est un grand réseau de données. C'est la maîtrise de ces données qui a donné à Google un pouvoir qui dépasse aujourd'hui l'internet. L'indexation des pages web et la pertinence des résultats renvoyés fait la qualité d'un moteur de recherche.

Au niveau de la structuration des pages web, de nouvelles balises ont été introduites avec le HTML 5. Le sémantisme associé à la structure a acquis une véritable autonomie. En effet, comme le présente le schéma ci-dessous, la balise générique à laquelle on associait des valeurs d'identifiant pour les spécifier a été remplacé par un nouveau jeu de balise dédié à la mise en page des sites web. Ainsi, un site web rédigé en HTML5 permet d'indiquer ce qui relève de l'information et ce qui relève de l'ergonomie, c'est-à-dire du moyen d'accéder à l'information ou de naviguer en son sein.

[14]



Il sera alors plus aisé de paramétrer un automate pour identifier les balises où se trouve l'information et l'extraire. Car l'enjeu du web sémantique est celui-ci : permettre à un

[14] Pour plus d'informations, consulter http://www.startyourdev.com/html/tag-html-balise-nav.

automate de lire des pages web comme le ferait un humain et sans intervention. Le HTML 5 permet de localiser l'information, mais pour la traiter, il s'est construit un jeu d'ontologies. Nous ne nous attarderons pas sur la syntaxe de création des ontologies car elles ne sont pas gérées directement par XQuery. Nous tenons toutefois à faire un rapprochement entre la constitution du thésaurus et celle de ressources pour l'exploitation du web sémantique, où des liens sémantiques sont créés entre les mots à travers des rapports hiérarchiques.

Dans la perspective du *Big Data*, il est important de pouvoir localiser l'information dans le but de l'extraire. Une structuration normalisée des pages web permet de penser une automatisation des requêtes sans développer de stratégie *ad hoc* pour chaque site. De plus, l'emploi d'ontologies permet de lier sémantiquement des unités lexicales et donc d'élargir le champ de recherche. Le web est devenu une ressource à ne pas négliger pour l'Humaniste numérique. Savoir l'exploiter au mieux permet une récolte de données plus volumineuse et plus pertinente et facilite la constitution d'un corpus de meilleure qualité.

2.5.3. La constitution automatique de corpus

La récolte de corpus textuel peut s'avérer fastidieuse, en particulier sur le web. Toutefois, XQuery peut rapidement offrir une solution permettant de créer un automate. Avant d'entrer dans le détail, il est nécessaire d'aborder certaines fonctions avancées d'XQuery que nous avons choisi de ne pas aborder précédemment dans un souci de clarté didactique.

XQuery dispose de nombreuses fonctions, mais il est également possible de créer ses propres fonctions. Pour cela, comme dans les langages procéduraux, il faut d'abord les déclarer avec un espace de nommage ou *namespace*, ici la fonction sera locale, ainsi que préciser les arguments et leurs types :

```
declare function local:getUrl($url as xs:string,$para as
xs:string)
```

Une fonction est un programme qui renvoie une valeur, ce qui explique l'omniprésence de l'instruction `return` dans un langage fonctionnel comme XQuery.

Les branchements conditionnels sont également disponibles en XQuery en posant la condition `if` et sa condition, `then`, introduit l'instruction à exécuter si la condition se vérifie et `else` celle qu'il faudra exécuter si ce n'est pas le cas.

Il existe des modules dédiés à la récupération de données sur le web. `http:request` récupère le contenu des balises de la page. `[1]` correspond à la balise `<head>` et `[2]` à `<body>`. C'est la seconde qui comportera le texte, la première étant consacrée aux métadonnées.

`http:parse` permet de spécifier le nom de la balise particulière à récupérer.

L'extraction d'informations d'un site se fait en combinant ces modules, créant un aspirateur entièrement paramétrable :

(:Déclaration de la fonction:)

```
declare function local:getUrl($url as xs:string,$para as
xs:string)
```

Si l'argument est une adresse web, l'automate lance la récupération dans la balise `<body>` du lien spécifié dans les balises dont le nom est donné en argument, sinon une erreur est renvoyée:

```
{
if(matches($url,"http://"))
  then
      for $x in (html:parse(http:send-request
    (<http:request method='get' override-media-
type='application/octet-stream'
href="{$url}"/>)[2])//*[name(.)=$para])
return $x
  else "Erreur rencontrée"
};
```

(:Déclaration des arguments:)

```
let
$url:="http://ec.europa.eu/contracts_grants/grants_en.htm"
let $mots :=("Aqua.*", "fish.*", "p.che")
```

(:La fonction string-join() renvoie la reconstruction sous forme de chaîne de caractère le contenu de balises. Ici, l'arborescence XML est la page web fournie dans la déclaration de la variable $url et la balise à extraire est <a> dont le contenu sera extrait s'il contient les mots-clés stockés dans la variable $mots:)

```
return
string-join
(local:getUrl(
  data(local:getUrl($url,"a")[(. contains text {$mots} any
word using wildcards)]/@href), "p")," ")
```

XML constitue une solution complète permettant d'agir à la fois sur la structure et la donnée. Cette convergence, qui est à l'origine de sa création, en fait l'outil adapté pour plusieurs applications, comme le traitement d'isotopies et l'alignement de corpus.

3. Perspectives applicatives

XML met en relation du texte, des unités lexicales et des métadonnées entre elles. Cette dernière partie aura pour objectif de présenter un champ de recherche qui dépasse le cadre de ce mémoire de Master et qui sera à exploiter dans le cadre d'un doctorat. XQuery apparaît comme la solution technologique la plus adaptée aux Humanités numériques et les perspectives applicatives sur lesquelles nous souhaitons nous concentrer sont de deux types.

La première application se fera dans le domaine de la sémantique. L'emploi du thésaurus constitue un puissant outil de mise en réseau sémantique. Une étude formelle de la sémantique et des isotopies sera un préalable à la création d'un thésaurus qui sera à écrire, pendant électronique du dictionnaire idéologique que Julio Casares a constitué pour

l'espagnol [15] . L'une des portées d'un tel dictionnaire serait de pouvoir mettre en correspondance des textes à travers leurs isotopies et mettre en avant des questions de genre.

La seconde constituera un apport à la traductologie. La mise en corespondance d'un texte et de sa traduction permettront de croiser les ressources et de procéder à un alignement de corpus.

3.1. La question de l'interprétation : sème, isotopie, corpus

Les répercussions liées à l'immixtion de l'outil informatique dans le domaine des sciences du langage sont nombreuses ; les différentes difficultés rencontrées par l'automate ont rendu concrètes la variabilité et la « déformabilité extrême » (A. Culioli) des unités lexicales, dont la stabilité intensive apparente – matérialisée par la productivité de l'activité lexicographique – se heurte à la classe indéfinie de valeurs qu'elles sont susceptibles de revêtir en discours. Ainsi, l'informatique remet en perspective le débat sur la nature des processus interprétatifs en questionnant sa computationnalité, sa détermination logique et sa systématicité ; par conséquent, elle interroge nécessairement d'une part les types d'unités qui entrent en compte dans l'interprétation d'une production linguistique (d'où proviennent-elles ? Dans quelle mesure sont-elles systémiques ?), d'autre part les relations qu'elles sont susceptibles d'entretenir entre elles. F. Rastier résume en ces termes : « que fait-on quand on lit un texte, et d'où provient le sentiment de son unité ? »[16].

Le succès des grammaires génératives repose sur l'idée, séduisante du point de vue de la validité et de la faisabilité applicatives de l'automate, que la production du texte – plus généralement, de toute activité verbale, ou plus encore communicationnelle – repose sur la dérivation d'engrammes cognitifs profonds, en nombre restreint et universels, s'incarnant dans des structures de surface dont les possibilités d'agencement formel, quoique potentiellement non définissables, sont virtuellement finies. Conversement, l'activité d'interprétation repose sur la « primitivation » des structures de surface et sur la résurgence des structures profondes.

[15] J. Casares, *Diccionario ideologico de la lengua española*. Barcelone : Gustavo Gili, 1990.
[16] F. Rastier, *Sémantique interprétative*. 3ème édition mise à jour et augmentée. Paris : Presses Universitaires de France, 2009 (1987), p. 9.

L'impossibilité de parvenir à réaliser automatiquement ou algorithmiquement le parcours bilatéral structures profondes-structures de surface a favorisé, comme nous l'avons déjà évoqué en 1., le développement d'appareils théoriques centrés sur l'agencement effectif du discours, ce dernier étant vu comme l'instanciation d'un certains nombre de traits formels. Parmi ces différents modèles, nous adoptons ici le point de vue de la sémantique interprétative de F. Rastier, elle-même issue d'une part de la sémantique structurale d'A. J. Greimas, d'autre part de l'analyse componentielle de B. Pottier.

Pour F. Rastier, la surface textuelle n'est compréhensible que comme s'insérant dans un contexte – et, plus réellement, dans un intertexte – donné. L'instanciation du jeu de traits sémantiques est déterminée par les données contextuelles (le terme devant nécessairement être compris dans un sens fort : données pragmatiques « classiques », mais aussi déterminants socio-culturels, cotexte antérieur, informations précédemment accumulées dans l'activité discursive, etc.), émetteur comme récepteur assumant que la prise en charge de l'appareil formel de l'expressivité y puise nécessairement son interprétabilité et son unité de sens. Ainsi, l'ambiguïté des unités lexicales ou des tournures syntaxiques, latente au niveau de la parole, est désamorcée au niveau du discours par leur lecture en creux de l'espace pragmatique qu'elles emportent. Au niveau le plus fondamental, la solidarité du discours réside dans ce que l'interprétant opère une recherche constante des indices de cette référence textuelle, de sorte que les unités du discours se trouvent virtuellement dans un rapport de toutes à toutes. Par la suite, la motivabilité pragmatique présumée des segments de discours trouve confirmation dans l'établissement de relations formelles, qui apparaissent ainsi *objectives*. Dans le cadre de la sémantique interprétative, ces relations sont appelées isotopies et se définissent comme des récurrences entre traits sémantiques :

> « fondamentalement, une isotopie est instituée par une série de relations d'identité entre sèmes. Ces relations induisent des relations d'équivalences entre sémèmes. Mais le modulo de l'équivalence n'est pas une donnée et il faut généralement parcourir des équivalences pour l'identifier. Cependant, plusieurs stratégies d'inférence peuvent être possibles ; et divers lecteurs obtiennent des scores différents en raison de la disparité de leurs connaissances encyclopédiques. Dans tous les cas, la description de l'isotopie est conditionnée par la compétence interprétative. Cela conduit à un déplacement de la problématique. En général, on considère l'isotopie comme une forme remarquable de combinatoire sémique, un effet de la combinaison des sèmes. Ici au contraire, où l'on procède paradoxalement à partir du texte pour aller vers ses éléments, l'isotopie apparaît comme un

> principe régulateur fondamental. Ce n'est pas la récurrence de sèmes déjà donnés qui constitue l'isotopie mais à l'inverse la présomption d'isotopie qui permet d'actualiser des sèmes, voir *les* sèmes. », F. Rastier, *ibid.*, p. 12

Si nous reviendrons par la suite plus en détail sur les notions de *sème* et d'*isotopie*, nous pouvons d'ores et déjà souligner que le contexte de production est consubtantiel du texte car il permet de satisfaire ce que, par exemple, D. Sperber et D. Wilson appellent la *présomption de pertinence* : toute activité verbale s'auto-justifie en assurant à ses interprètes qu'il porte plus que son seul agencement systémique – sans quoi *il fait froid ici* signifierait immuablement *il fait froid ici* et jamais *ferme la fenêtre*, que *carré* signifierait toujours *carré* et que, plus généralement, le sens serait toujours essentiellement référentiel – et qu'il est possible, en associant l'expression systémique à ses conditions de production, d'accéder au sens visé.

Nous étudierons brièvement les conséquences de cette approche au niveau de la notion contexte, puis présenterons formellement les notions de *sème* et d'*isotopie* au sein de ce modèle.

3.1.1. Du mot au corpus et du texte au contexte

La sémantique interprétative prend le contre-pied exact des approches génératives : le sens ne se définit pas par référence à des structures précodées internes à la cognition mais par référence à un ensemble de données variable, externes au locuteur et qui ne définissent pas la langue elle-même. Plus précisément, un mot n'a de sens que pris dans un réseau par lequel il manifeste formellement sa référence au contexte qui l'active localement. Ces réseaux, dont nous avons vu plus haut qu'ils correspondent à la « responsabilité interprétative » de l'émetteur aussi bien que du récepteur, définissent un matériel sémantique qui se superpose à l'orientation pragmatique du texte qui les contient, structurent ce dernier et le positionnent dans la continuité ou en rupture d'autres textes (question de la généricité et de l'intertextualité). F. Rastier résume ainsi sa position :

> « (i) le sens est un niveau d'objectivité qui n'est réductible ni à la référence, ni aux représentations mentales. Il est analysable en traits sémantiques qui

> sont des moments stabilisés dans des parcours d'interprétation. (ii) La typologie des signes dépend de la typologie des parcours dont ils sont l'objet. (iii) Le sens est fait de différences perçues et qualifiées dans des pratiques. C'est une propriété des textes et non des signes isolés (qui n'ont pas d'existence empirique). (iv) Le sens d'une unité est déterminé par son contexte. Le contexte, c'est tout le texte : la microsémantique dépend donc de la macrosémantique. (v) Les unités textuelles élémentaires ne sont pas des mots mais des *passages*. Un passage a pour expression un *extrait* et pour contenu un *fragment*. (vi) Sur le plan sémantique, les traits pertinents sont organisés pour composer des *formes sémantiques*, comme les thèmes, qui se détachent sur des *fonds sémantiques*, les isotopies notamment. Les formes sémantiques sont des moments stabilisés dans des séries de transformations, tant au sein du texte qu'entre textes. »
> « (i) Si le morphème est l'unité linguistique élémentaire, le texte est l'unité minimale d'analyse, car le global détermine le local. (ii) Tout texte procède d'un genre qui détermine sans les contraindre ses modes génétique, mimétique et herméneutique. (iii) Par son genre, chaque texte se relie à un discours. (iv) Tout texte doit être rapporté à un corpus pour être interprété. (v) Le corpus préférentiel d'un texte est composé de textes du même genre. Les parcours interprétatifs au sein du texte sont inséparables des parcours interprétatifs dans l'intertexte nécessaire que constitue le corpus. », *ibid.*, préface à la troisième édition, p. VI.

La notion de contexte est donc considérablement élargie – ne serait-ce parce qu'elle contient intrinsèquement celle de *cotexte* et d'*intertexte* – et devient le noyau sémantique du texte (« sans adopter une perspective générative, on dira que le contenu d'un texte ne peut être décrit qu'en fonction de données pragmatiques », *ibid.*, p. 34). En ce sens, il est légitime de se demander si un texte a réellement un sens du point de vue de sa seule dénotation ; par l'absurde, Rastier montre, en recourant à l'exemple d'un portrait d'Hadrien tiré des *Carmina Sybillina*, que la neutralisation des données socioculturelles fragmente l'unité de sens du texte : les commentateurs contemporains hésitent entre deux interprétations radicalement opposées du passage (éloge ou critique), car il y a tout un continuum de matérialité expressive qui se trouve perdu et inaccessible (effets de certaines tournures syntaxiques, stéréotypie des unités lexicales utilisées, contexte historique et politique particuliers, etc.). Le produit verbal ne contient donc pas formellement son message, en ce sens qu'il n'aurait qu'à être décodé par une composante pragmatique cognitive. Pour autant, il ne s'agit pas de dire qu'il devient ininterprétable quand on l'éloigne de son contexte de production, mais que lorsque le contexte devient inaccessible, la probabilité d'accès à l'intention interprétative s'éloigne d'autant, de sorte que les interprétations se multiplient et que le texte perd sa solidarité. Ainsi, c'est bien littéralement le contexte, le global, qui univocise et surdétermine le local, le produit verbal ponctuellement formé.

3.1.2. L'unité de l'analyse sémantique : le sème

Les unités du texte ne sélectionnent donc pas des portions variables du contexte ; au contraire, c'est immuablement la nécessité d'enrichir le strict plan du dit par la prise en compte du contexte qui active dans le matériau formel – unités lexicales, masses syntaxiques, procédés stylistiques, etc. – des traits sémantiques susceptibles de s'organiser en réseaux et d'opérer la ligature texte-contexte.

Ces traits sémantiques sont appelés *sèmes* et, dans l'héritage structural, se définissent comme des marques distinctives étayant la stabilité apparente des unités du système et participant de leur illusion référentielle. Par exemple, dans le fragment *l'homme est une femme comme les autres*, *homme* se distingue de *femme* par le sème *genre*. Un ensemble de sèmes formant un terminal intensif (*homme*, *femme*) est appelé un *sémème* ; un ensemble de sémèmes définis par la récurrence d'un ou plusieurs sèmes, un *taxème*.

Le sème n'est pas une unité spécifique de la sémantique interprétative ; néanmoins, conséquence de la dimension pragmatique consubstantielle du sens au sein de ce modèle, il y acquiert une définition essentiellement utilitariste : « "le *sème* est le trait sémantique distinctif d'un sémème, relativement à un petit ensemble de termes réellement disponibles et vraisemblablement utilisables chez le locuteur dans une circonstance donnée de communication" (Pottier, 1980a, p. 169). Elle ne peut que chagriner les tenants d'une sémantique universaliste *a priori* : l'identification d'un sème dépend on le voit de cinq conditions hiérarchisées, toutes dépendantes du contexte linguistique et/ou de l'entour pragmatique. » (*ibid.*, p. 33). La dimension contexuelle du sème ne signifie pas qu'aucune convention linguistique n'intervient dans sa définition ; néanmoins, seul le contexte déterminera si d'une part un sème conventionnel sera, en discours, effectivement activé ou au contraire neutralisé, d'autre part si un sème non conventionnel se greffera sur un ou plusieurs sémèmes.

La présence des sèmes dépend donc d'un équilibre entre conventions linguistiques mutuellement présentes à l'esprit des interactants et dimension prescriptrice du discours. Ils se définissent ainsi selon deux axes d'opposition : l'axe générique/spécifique et l'axe inhérent/afférent. Un sème est dit générique quand il factorise l'ensemble des sémèmes d'un taxème (par exemple, *à roues* dans la classe des *voitures*) ; il est dit spécifique quand il

permet de distinguer un sémème des co-sémèmes d'un taxème (*avec capote* pour *décapotable*, par rapport à *coupé*, *berline*, etc.). Par ailleurs, un sème est dit inhérent quand il détermine des sémèmes d'un même taxème (qu'il soit générique ou spécifique, factorisant ou distinctif) ; il est dit afférent quand il installe des relations inférentielles, « asymétriques et non réflexives » (*ibid.*, p. 54) entre sémèmes de taxèmes différents. Relativement à ce dernier cas, l'association culturocentrée des sémèmes *faiblesse*/*femme* et *force*/*homme* (l'exemple est de Rastier) est un cas de relation afférente. Ainsi, les sèmes étant des traits distinctifs ou spécifiant, ils désignent toujours des relations entres sémèmes, et incidemment des rapports entre éléments d'une même classe (qu'est-ce qui les rassemble au pont de constituer la solidarité de la classe ? Qu'est-ce qui les distingue au sein de la classe de sorte qu'elle n'est pas réduite à un élément?) et entre éléments de classes différentes (comment se font les passages d'une classe à l'autre en discours ? Qu'est-ce qui explique la variabilité et l'instabilité des conventions lexicales?).

3.1.3. La notion d'isotopie

F. Rastier définit l'isotopie ainsi : « on appelle isotopie toute itération d'une unité linguistique. […] sur le plan du contenu, l'isotopie n'est pas définie exclusivement par la récurrence de classèmes, mais par celle de toutes les unités sémantiques, et donc aussi par celle des sèmes spécifiques » (*ibid.*, p. 92). L'isotopie est donc une structure dont les bornes ne sont pas fixées *a priori*, et qui se composent d'unités potentiellement minimales liant les sémèmes les uns aux autres.

La conséquence la plus directe de cette notion sur l'appareil linguistique traditionnel est de remettre en cause l'intégrité de la phrase, non du moins en tant que structure syntaxique ou typographique mais en tant que structure sémantique. Le sens d'un texte ne se laisse pas décrire comme un ensemble linéaire de phrases mais comme un ensemble d'isotopies dont les bornes excède les répartitions syntaxiques particulière – ce qui, là encore, n'interdit pas que la syntaxe participe de la constitution des isotopies.

L'isotopie est ainsi un principe qui assure la solidarité interne du texte et définit dans une certaine mesure son horizon de réception. Si l'isotopie repose sur les relations de

continuités entres sémèmes sur la base de leur itération sémique, et si les sèmes d'un textes sont par définition entre nombre fini, alors les isotopies sont elles-mêmes en nombre fini (nécessairement inférieur, d'ailleurs, à l'ensemble des sèmes) et hiérarchisables en fonction du nombre de sèmes qu'elles impliquent et de sémèmes qu'elles relient. De la sorte, les isotopies constituent des mécanismes de cohésion, et les possibilités de poly-isotopie sont tranchées par la prise en charge de l'acte de lecture par le récepteur (on en revient alors à l'inscription du texte dans son contexte de production et dans sa relation avec les autres textes qui partagent un déploiement isotopique similaire).

3.2. Perspectives applicatives : XML et le(s) texte(s)

3.2.1 Une ressource numérique entre fragments et texte : le thésaurus

Les mots ne sont donc pas indépendants les uns des autres. Dans un texte, les mots entrent en résonance dans un réseau sémantique qui va le structurer. Il ne sera pas difficile de retrouver dans un texte sur l'opéra des mots comme *aria*, *cantatrice*, *oratorio*. Ainsi, dans un texte sur l'informatique, des connexions vont se créer pour mettre en relation les mots. *Clavier*, *souris*, *puce* et *port* sont tous polysémiques. *Clavier* peut renvoyer à un instrument de musique, *puce* à un parasite, *souris* à un rongeur et *port* à une infrastructure maritime. Toutefois, c'est leur concomitance qui va activer un de leurs sens plutôt qu'un autre : chaque mot est un nœud de l'isotopie sémantique qui contribue à définir un texte. Celui-ci sera reconnu par un lecteur comme traitant de l'informatique sans même que le mot informatique ne soit employé une seule fois.

Julio Casares, dans son dictionnaire idéologique[17], met en relation des unités lexicales non pas en fonction de leur morphologie ou de leur définitions, mais de leur sens. Ainsi, des réseaux isotopiques peuvent être créés dans un dictionnaire. Le travail de Casares recense un volume conséquent de données. Publier un équivalent pour le français est un objectif à fixer pour une recherche approfondie dans le cadre d'une thèse de doctorat.

[17] *Ibid*.

Une fois établis, la comparaison des réseaux isotopiques rend possible le rapprochements entre des textes. Un texte pourra être considéré comme synonyme d'un autre texte si, comme au niveau de l'unité lexicale, ils partagent un certain nombre de traits sémantiques et donc d'isotopies. La mise en réseau d'isotopies sémantiques permet de relier des textes là où l'analyse lexicale de surface peut opposer un texte traitant de la sortie d'une nouvelle souris d'ordinateur et un autre évoquant celle d'un clavier tout en rapprochant le premier d'un texte rapportant une invasion de rongeurs.

En sus, un texte ne se définit pas simplement par le lexique employé mais aussi par son genre. Dans un quotidien, tous les articles de presse traitent de sujets différents. Ils ont pourtant tous en commun une structure argumentative caractéristique du texte journalistique qui constitue une signature textuelle. Ce sont ces éléments qui permettent au lecteur d'établir ce qui correspond à une synonymie, non plus uniquement lexicale mais également textuelle. La conjonction de ces éléments permet d'identifier ce que nous appellerons topométrie lexicale et qui justifie la cohérence d'un corpus.

Cette analyse linguistique va au-delà de la langue et peut être étendue dans une optique d'exploitation de synonymie interlingue. Le réseau d'isotopies sémantiques est constitué de liens de sens, non pas d'unités lexicales. C'est ce niveau d'abstraction qui rend la traduction possible d'une langue à l'autre. Au niveau lexical, la mise en correspondance d'isotopies va, entre autres, orienter un choix de traduction vers une unité lexicale ou une autre. Au niveau du texte, la traduction ne se fait plus au niveau des unités lexicales et des liens qu'ils entretiennent les uns avec les autres, mais par l'étude des réseaux isotopiques et de leurs liens idéologiques, transposables d'une langue à une autre.

XML nous semble être la meilleure solution technologique pour entreprendre une telle tâche. En effet, comme nous l'avons détaillé, les modules plein-texte permettent de faire une véritable analyse linguistique. De plus, le thésaurus proposé en natif est la structure idoine à cette problématique. La capacité de XQuery à opérer une gestion multilingue du texte grâce à des modules plein-texte dédiés est également un atout non négligeable dans ce projet. Enfin, XML est ce qui est aujourd'hui ce qui est le plus adapté au balisage, car c'est ce pour quoi il a été conçu : traiter de la donnée textuelle et la baliser.

3.2.2. Le croisement des structures : XPath pour l'alignement de corpus

Un texte n'est pas un flux de caractères, il s'agit d'un agencement ordonné d'informations. La structure du texte est aussi importante que les mots qui le composent. L'éditeur d'un texte bénéficie d'une certaine latitude quant à la mise en forme. La police de caractères, les marges, la pagination sont autant de choix qui forment d'une certaine manière la partie instable du texte. Il est rare qu'une page d'un livre de deux éditions différentes renvoient au même passage du texte. Le seul moyen de faire jouer une correspondance entre deux éditions ou un texte et sa traduction est de se servir de la structure profonde du texte. Il est plus simple de renvoyer à un acte et une scène qu'à un numéro de page. La structure profonde d'une pièce de théâtre comme *le Cid* est difficile à appréhender pour un humain. En effet, il est très difficile d'aller plus en profondeur que les deux premiers niveaux que nous venons d'évoquer. En revanche, il est aisé pour un automate de naviguer à travers un grand volume de données. XPath offre un moyen très efficace de tirer profit de l'adressage de la donnée pour l'extraire.

Les annexes 4, 5 et 6 sont trois versions d'un même texte, la *Déclaration universelle des droits de l'Homme* en français, en anglais et en chinois. Bien qu'extraites du même site web, elles présentent des mises en page différentes ne permettant pas un traitement informatique brut. Un marquage du texte reflétant celui-ci sera la clé d'une correspondance commune à toutes les versions. Une possibilité d'étiquetage serait la suivante :

```
<declaration>
  <preambule>
    <considerant>
      <phrase/>
    </considerant>
  </preambule>
  <article>
    <alinea>
      <phrase/>
    </alinea>
  </article>
</declaration>
```

L'accès à une donnée particulière se fait grâce à son chemin XPath. Considérons l'exemple suivant

```
<declaration>
```

```
  <article id="23">

    <alinea id="1">
      <phrase> Toute personne a droit au travail, au libre
               choix de son travail, à des conditions
               équitables et satisfaisantes de travail et à
               la protection contre le chômage.
      </phrase>
    </alinea>

    <alinea id="2">
      <phrase> Tous ont droit, sans aucune discrimination, à
               un salaire égal pour un travail égal.
      </phrase>
    </alinea>

    <alinea id="3">
      <phrase> Quiconque travaille a droit à une rémunération
               équitable et satisfaisante lui assurant ainsi
               qu'à sa famille une existence conforme à la
               dignité humaine et complétée, s'il y a lieu,
               par tous autres moyens de protection sociale.
      </phrase>
    </alinea>

    <alinea id="4">
      <phrase> Toute personne a le droit de fonder avec
               d'autres des syndicats et de s'affilier à des
               syndicats pour la défense de ses intérêts.
      </phrase>
    </alinea>

  </article>

  <article id="24">
    <alinea id="1">
      <phrase> Toute personne a droit au repos et aux loisirs
               et notamment à une limitation raisonnable de
               la durée du travail et à des congés payés
               périodiques.
      </phrase>
    </alinea>
  </article>

</declaration>
```

Il est désormais facile d'extraire une donnée. Si l'on veut connaître l'article 24, il suffit de faire un requête sur ce chemin XPath :

```
for $x in declaration/article[./@id=”24”]/alinea

return <article>{data($x)}</article>
```

ce qui renverra le résultat suivant :

> <article> Toute personne a droit au repos et aux loisirs et notamment à une limitation raisonnable de la durée du travail et à des congés payés périodiques. </article>

La requête peut mettre directement en correspondance les différentes traductions en gérant plusieurs bases de données, en cherchant l’article 23 alinéa 3 en français, anglais et chinois :

```
let $article := "23"

let $alinea := "3"

let $langue := ('fr','en','zh')

return <racine> {


for $l in $langue, $x in
db:open($l)/declaration/article[./@id=$article]/alinea[./@id=
$alinea]/phrase
return <article lang="{$l}">{data($x)}</article>

        } </racine>
```

Le résultat obtenu est un alignement direct des traductions du même texte :

> <racine>
>
> <article lang="fr">Quiconque travaille a droit à une rémunération équitable et satisfaisante lui assurant ainsi qu'à sa famille une existence conforme à la dignité humaine et complétée, s'il y a lieu, par tous autres moyens de protection sociale.</article>
>
> <article lang="en">Everyone who works has the right to just and favourable remuneration ensuring for himself and his family an existence

worthy of human dignity, and supplemented, if necessary, by other means of social protection.</article>

<article lang="zh">每一个工作的人，有权享受公正和合适的报酬，保证使他本人和家属有一个符合人的生活条件，必要时并辅以其他方式的社会保障。</article>

</racine>

Le formalisme XML autorise donc d'imaginer une telle application sur le web. En effet, la requête peut tout à fait créer une page web à la volée, en recréant une arborescence qui correspondrait à un page XHTML ou HTML 5, en insérant à la place de la balise <racine> une racine <html><head/><body>. Le contenu des variables pourra être spécifié par l'utilisateur au moyen d'une page PHP pour rendre l'utilisation dynamique.

Une autre perspective envisagée est de ne plus s'appuyer uniquement sur la structure du texte, mais également sur les réseaux sémantiques crées à l'aide du dictionnaire idéologique. Une fois un texte réduit à ses structures et catégorisé, les ponts d'une langue à une autre sont plus simples à construire. En effet, la recherche ne se fera plus sur des unités lexicales, mais des idées, réduisant ainsi les ambiguïtés dues à la polysémie. Il ne s'agira plus de traduire un mot, mais le sens derrière celui-ci. Le croisement du réseau et de la structure permettra de s'approcher d'un alignement de corpus fin.

Cette opération pourra se faire à la volée en s'appuyant une fois encore sur le formalisme XML. Les technologies web permettent de réaliser un travail d'éditeur. Le CSS 3[18] met en forme les pages web en fonction des balises et des attributs, rendant possible l'affichage simultané du texte en langue source et en langue cible. La composition sera la dernière touche d'un processus de transformation qui aura décodé le texte pour le réduire à sa structure et un flux de données pour ensuite le faire revivre sous une forme à la fois nouvelle et familière.

[18] Pour approfondir la syntaxe et les options proposées par le CSS 3 : D. McFarland, *CSS3: The Missing Manual*. Sebastopol, CA : O'Reilly, 2012.

Conclusion

Les Humanités numériques ont des problématiques communes, et l'exploitation du texte en est une. Le développement des ressources numériques est en constante progression, mais il devient difficile d'appréhender avec efficacité cette nouvelle manne de données. La linguistique et la linguistique informatique en particulier ont œuvré pour mettre en place des solutions adaptées. Il existe de nombreux langages de programmation avec leurs spécialités, leurs spécificités et leurs points faibles. De nombreux outils ont également été développés, certains pouvant etre intégrés dans une chaîne de traitement, d'autres pas. XML et XQuery offrent des possibilités qui permettent déjà une exploitation du texte à plusieurs niveaux de profondeur.

Le premier objectif de ce travail est de rendre accessible à l'Humaniste numérique des moyens technologiques pour créer ses propres ressources numériques de manière éclairée, à la fois sur le *comment* mais surtout le *pourquoi*. Il nous est apparu essentiel de ne pas tomber dans l'écueil des luttes de chapelles ; c'est pourquoi il est important de remettre XML dans son contexte. Ce formalisme est rigoureux, ce qui peut rebuter certains. Si PERL a su séduire des linguistes par la facilité avec laquelle on peut obtenir des résultats, qu'ils soient justes ou faux d'ailleurs, il est important que l'Humaniste numérique ne fasse pas les mêmes erreurs et se dote des outils les plus adaptés à son exploitation du texte. Car son objet de travail est bel et bien le texte.

XML et le texte sont au cœur de cette recherche, qui se projette au-delà de cet objet. Le second objectif de ce mémoire est d'identifier des problématiques liées à la détection de réseaux d'isotopies ainsi que trouver les solutions technologiques en vue de leur analyse dans le cadre d'un projet de recherche sur la synonymie textuelle.

Bibliographie

- Abeillé, Anne. *Les nouvelles syntaxes, grammaires d'unifications et analyse du français*. Paris : Armand Colin, 1993.
- Abney, Steven. *Parsing by Chunks.* In : R. Berwick, S. Abney & C. Tenny (eds.), « Principle-Based Parsing ». Dordrecht : Kluwer Academic Publishers, 1991.
- Aït-Mokhtar, Salah, Chanod, Jean-Pierre, Roux, Claude. « Robustness beyond Shallowness: Incremental Deep Parsing » In : *Natural Language Engineering 8*. Cambridge : Cambridge University Press, 2002.
- Benveniste, Emile. *Problèmes de linguistique générale*. París : Gallimard, 1974.
- Boitet, Christian. *Bernard Vauquois et la TAO : 25 ans de traduction automatique : analectes*. Gières : Association Champollion, 1989.
- Bray, Tim, Paoli, Jean, Sperberg-Mcqueen, C. Michael, *et al*. Extensible markup language (XML). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, 1998.
- Buvet, Pierre-André, Cartier, Emmanuel, Issac, Fabrice, Mathieu-Colas, Michel, Mejri, Salah & Madlouni, Yassine. « Morfetik, ressource lexicale pour le TAL ». In : *Actes de TALN 2009*, Senlis, 2009.
- Casares, Julio. *Diccionario ideologico de la lengua española*. Barcelone : Gustavo Gili, 1990.
- Doualan, Gaëlle, Boucher, Mathieu, Brixtel, Romain, *et al.* « Détection de mots-clés par approches au grain caractère et au grain mot ». *Actes de l'atelier de clôture du huitième défi fouille de texte (DEFT)*, 2012, p. 45-52.
- François, Jacques (éd.). *Théories contemporaines du changement sémantique, Mémoires de la société de linguistique de Paris*, 2000, t. IX, Louvain : Peeters..
- Francopoulo, Gil.*LMF Lexical Markup Framework*. Londres : ISTE, 2013.
- Friedl, Jeffrey. *Mastering regular expressions*. Sebastopol, CA : O'Reilly Media, Inc., 2006.
- Habert, Benoit, Fabre, Cécile, Issac, Fabrice. *De l'écrit au numérique. Constituer, normaliser et exploiter les corpus électroniques*. Paris : Masson, 1998.

- Issac, Fabrice & Salvador, Xavier-Laurent. *La synonymie médiévale en graphes*. Paris : Université Paris-Sorbonne, février 2012.

- Issac, Fabrice. « Place des ressources lexicales dans l'étiquetage morphosyntaxique », in *L'information grammaticale*, n°122, 2009, pp. 10–18.

- Jeanneret Yves, Meeùs Nicolas, Soutet, Olivier, *et al.*. *Que faisons-nous du texte ?*. Paris : Presses de l'Université Paris-Sorbonne, 2012.

- McFarland, David Sawyer. *CSS3: The Missing Manual*. Sebastopol, CA :O'Reilly, 2012.

- Molinié, Georges. *Éléments de stylistique française*. Paris : Presses Universitaires de France, 2011, 4e édition.

- Piotrowski, Rajmund, Romanov, Yourij. « Machine Translation in the former Soviet Union and in the Newly Independent States (NIS) », in *Histoire Épistémologie Langage*, Tome 21, fascicule 1, 1999.

- Poibeau, Thierry. *Du texte brut au Web sémantique*. Paris : Hermès, 2003.

- Rastier, François. «Sémantique du web vs. Semantic Web ? Le problème de la pertinence» in *Syntaxe & Sémantique 9*. Caen : Presses universitaires de Caen, 2008.

- Rastier, François. *La mesure et le grain*. Paris : Honoré Champion, 2011.

- Rastier, François. *Sémantique interprétative*. 3ème édition mise à jour et augmentée. Paris : Presses Universitaires de France, 2009 (1987).

- Rastier, François. *Textes et sens*. Paris : Didier Érudition, 1996.

- Rastier, François.« Doxa et sémantique de corpus ». In :*Langages 2*, 2008.

- Ray, Erik. *Learning XML*. Sebastopol, CA : O'Reilly, 2002.

- Schmidt, Helmut. "Probabilistic Part-of-speech Tagging Using Decision Tree". IMS-CL, Universität Stuttgart.

- Silberztein, Max « Le dictionnaire électronique des mots composés ». In: *Langue française*, Vol. 87, pp. 71-83.

- Van Assem, Mark, Gangemi, Aldo & Schreiber, Guus. « Conversion of WordNet to a standard RDF/OWL representation». In : *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Gênes, Italie, 2006.

- Vergne, Jacques. « Analyse syntaxique automatique de langues : du combinatoire au calculatoire », in TALN'2001.

- Vergne, Jacques. « Découverte locale des mots vides dans des corpus bruts de langues inconnues, sans aucune ressource », in *JADT 2004 : 7es Journées internationales*

d'Analyse statistique des Données Textuelles, 2004.

- Vergne, Jacques. « Un outil d'extraction terminologique endogène et multilingue », in *TALN 2003*, Batz-sur-Mer, 2003.

Annexes

Annexe 1

Page d'accueil du site http://www.isilex.fr

```
<html>
	<head>
	  <meta charset="utf-8"/>
	  <title>Isilex.fr</title>
	  <link rel="stylesheet" type="text/css" href="CSS/style.css"/>
	  <link rel="icon" type="image/png"
	href="http://www.crealscience.fr/images/Anonymous_keyring.png" />
	  <link
	href='http://fonts.googleapis.com/css?family=Belleza|Nixie+One|Open+Sans|PT+Serif+Caption|Eagle+Lake&subset=latin,latin-ext'
	rel='stylesheet' type='text/css'>
	</head>

	<body>
	  <div id='top'>
		  <div id='titrepos1'><li><a href='http://www-ldi.univ-paris13.fr'><img src='http://www-ldi.univ-paris13.fr/old-ldi/images/M_images/logo-ldi.jpg' width="80"
	height="30"></a></li>Isilex, Easy Lexical Ressources Portal </div>
	  </div>
		  <div id='global'>
		  <div id='menu'>
		  <a href='http://www.isilex.fr'>Home</a>
		  <a href='http://test.isilex.fr'>See ISILEX</a>
		  <a href='http://www.crealscience.fr/ModifierDemo.php'>Try
	ISILEX</a>
		  <a href='http://www.isilex.fr/links.php'>Links</a>
		  <a href='http://www.isilex.fr/download.php'>Download</a>
		  <a href='http://www.isilex.fr/contacts.php'>Contact Team</a>
		  <a href='http://www.isilex.fr/lexichamp'>Isotopies
	Automatiques</a>
		  </div>
		  <div id='contenu'>
			  <div id='fondaffcih'>
				  <div id='affichdata'>
					  <div class='texte'>
	<h2>What is this ?</h2>
	<p><a href='http://www.crealscience.fr/ModifierDemo.php'>Isilex</a>
	is an XML, collaborative or stand alone, Dictionnary Tool
	builder.</p>
	<p>You can see an automatic application for automatic Detection of
	Isotopies thru Litterary texts <a
	href='http://www.isilex.fr/lexichamp'>here</a></p>
```

```
<p>You can play with the complete editor <a
href='http://www.crealscience.fr/ModifierDemo.php'>here</a></p>
<p>It is a Tool for <b>consulting</b>, <b>editing</b> and
<b>publishing</b> XML Lexicographic TEI Compliant Datas.</p> See <a
href='htt://test.isilex.fr'>here</a>.
<p>Linguists teams obviously need time for working on datas, and
can't spend time on self-learning technical solutions for
Lexicographic datas editioning.</p>
<p>On an other hand, <a href=''>XML</a> and <a href=''>TEI</a> are
new and efficient standards for international datas development. But
Learning conventions, methods, programing solutions is not quite easy
to learn for someone not coming from numeric literacies. Working on
will soon be a nightmare for collaborative work.</p>

<h2>Solution ?</h2>
We do offer a solution for creation, edition <b>and Web hosting</b>.
<a href='http://www.isilex.fr'>IsiLex</a> - standing for <a
href='http://en.wikipedia.org/wiki/Isidore_of_Seville'>Isidorus</a>
Lexicograph - is a Lexicographic CMS (Content Management System).
It's a specific and easy solution for lexical ressources creations
developed in <a href='http://www-ldi.univ-paris13.fr'>LDI CNRS
Laboratory</a> by Researchers coming from European Universities.
Isilex makes available an easy way for your teams to produce TEI
compliant XML Ressources and Databases without initial formation.
It's an integrated XML TEI compliant WYSIWYG solution. First of all
it's <b>free</b>. The core of the editing project is an harmonious
organization of separate tools already existing pull inside a Web CMS
dedicated to dictionnary building with <li id='oldst'><a
href='http://Basex.org'>BaseX</a> from Konstanz University in it</li>
</p>

<h2>What else ?</h2>
<p>It's a <b>Web Hosting solution for lexicography</b> with full
integration, help for deployment, XML Database Assistance. You will
have full ftp access and a domain name like
<b>yourdictionnary.isilex.fr</b> </p>


<h2>Our goal ?</h2>
<p>
The project is born three years ago, while working on <a
href='http://www.crealscience.fr'>The Dictionnary of the medieval
Sciences</a>. We decided to produce a tool dedicated to production,
edition, consultation and XML datas complying solutions with other
international projects and easy to use, even for People with no time
to spend in XML sharewares solutions like <a
href='www.oxygenxml.com/'>Oxygen</a>.
</p>

<p><a href=''>Isilex</a> could be an easy standalone or collaborative
solution, focused on XML producing datas. In a few clics, you can
read, append, alter or delete XML Datas without seing even one
balise, everything done in a famous wysiwig editor.</p>

<h2>How To ?</h2>
<p>
You build a lexicographic project, based on the type of ressources
you aim at producing (monolingual, bilingual).)
<ul>
   <li id="oldst"> You build a structure</li>
   <li id="oldst"> Your teams can collaborate.</li>
```

```
         <li id="oldst"> You moderate.</li>
         <li id="oldst"> You can edit a website or a wondeful PDF
      book.</li>
      </ul>
      </p>
      <h2>Is there anyone using it ?</h2>
      <p>Yes, indeed: <a
      href='http://crealscience.isilex.fr'>crealscience.isilex.fr</a>.</p>
      <p>Another project, by the same author: <a
      href='http://isilex.biblehistoriale.fr'>biblehistoriale.isilex.fr</a>
      </p>
      <p>Automatic detection of Lexical Isotopies <a
      href='http://www.isilex.fr/lexichamp'>Here</a></p>
      <p><a>I-Def</a> Society</p>


                                    </div>
                              </div>
                                    <div id='bas'>
      Powered by <a href=''>Isilex</a>, a Lexicographic CMS <br/>
      XLS, FI, GP, MF, JE, YS<a
      href="https://plus.google.com/103783772087446954534"
      rel="publisher">Retrouvez-nous sur Google+</a>
                                    </div>
                        </div>
                  </div>
      </body>
</html>
```

Annexe 2

DTD établie pour l'évaluation des perfomances du moteur EIBM dans le cadre du projet collectif de la promotion 2013/14 du Master Professionnel TILDE.

```
<!ELEMENT racine (meta|texte)+>
<!ELEMENT meta (site|url|registre|langue)+ >
<!ELEMENT site (#PCDATA)* >
<!ELEMENT url (#PCDATA)* >
<!ELEMENT registre (#PCDATA)* >
<!ELEMENT langue (#PCDATA)* >
<!ELEMENT texte
(#PCDATA|titre|maladie|medicament|therapie|EPIDEMIE|TRAITEMENT
|ALERTE|PATHOLOGIE|BRUIT)* >
<!ELEMENT titre
(#PCDATA|maladie|medicament|therapie|EPIDEMIE|TRAITEMENT|ALERT
E|PATHOLOGIE|BRUIT)* >
<!ELEMENT maladie
(#PCDATA|medicament|therapie|EPIDEMIE|TRAITEMENT|ALERTE|PATHOL
OGIE|BRUIT)* >
```

```
<!ELEMENT medicament
(#PCDATA|maladie|therapie|EPIDEMIE|TRAITEMENT|ALERTE|PATHOLOGI
E|BRUIT)* >
<!ELEMENT therapie
(#PCDATA|maladie|medicament|EPIDEMIE|TRAITEMENT|ALERTE|PATHOLO
GIE|BRUIT)* >
<!ELEMENT EPIDEMIE
(#PCDATA|maladie|medicament|therapie|TRAITEMENT|ALERTE|PATHOLO
GIE|BRUIT)* >
<!ELEMENT TRAITEMENT
(#PCDATA|maladie|medicament|therapie|EPIDEMIE|ALERTE|PATHOLOGI
E|BRUIT)* >
<!ELEMENT ALERTE
(#PCDATA|maladie|medicament|therapie|EPIDEMIE|TRAITEMENT|PATHO
LOGIE|BRUIT)* >
<!ELEMENT PATHOLOGIE
(#PCDATA|maladie|medicament|therapie|EPIDEMIE|TRAITEMENT|ALERT
E)* >
<!ELEMENT BRUIT
(#PCDATA|maladie|medicament|therapie|EPIDEMIE|TRAITEMENT|ALERT
E|PATHOLOGIE)* >
<!ATTLIST maladie eval (précis|bruit|silence) #REQUIRED
          lm (non) #IMPLIED >
<!ATTLIST medicament eval (précis|bruit|silence) #REQUIRED
          lm (non) #IMPLIED >
<!ATTLIST therapie eval (précis|bruit|silence) #REQUIRED
          lm (non) #IMPLIED >
<!ATTLIST EPIDEMIE eval (précis|bruit|silence) #REQUIRED
          inex (oui) #IMPLIED >
<!ATTLIST TRAITEMENT eval (précis|bruit|silence) #REQUIRED
          inex (oui) #IMPLIED >
<!ATTLIST ALERTE eval (précis|bruit|silence) #REQUIRED
          inex (oui) #IMPLIED >
<!ATTLIST PATHOLOGIE eval (précis|bruit|silence) #REQUIRED
          inex (oui) #IMPLIED >
<!ATTLIST BRUIT eval (précis|bruit|silence) #REQUIRED >
```

Annexe 3

Schema XML du thesaurus pour XQuery.

```
<xs:schema
targetNamespace="http://www.w3.org/2007/xqftts/thesaurus"
elementFormDefault="qualified"><!--
        equivalence relationships (synonyms): PREFERRED TERM (USE),
NONPREFERRED USED FOR TERM (UF);

        hierarchical relationships: BROADER TERM (BT), NARROWER
```

TERM (NT), BROADER TERM GENERIC (BTG), NARROWER TERM GENERIC (NTG), BROADER TERM PARTITIVE (BTP), NARROWER TERM PARTITIVE (NTP), TOP Terms (TT); and

associative relationships: RELATED TERM (RT).
--><xs:simpleType name="defined-relationship"><xs:restriction base="xs:token"><xs:enumeration value="USE"/><xs:enumeration value="UF"/><xs:enumeration value="BT"/><xs:enumeration value="NT"/><xs:enumeration value="BTG"/><xs:enumeration value="NTG"/><xs:enumeration value="BTP"/><xs:enumeration value="NTP"/><xs:enumeration value="TT"/><xs:enumeration value="RT"/></xs:restriction></xs:simpleType><xs:simpleType name="relationship"><xs:union memberTypes="defined-relationship xs:token"/></xs:simpleType><xs:complexType name="synonym"><xs:sequence><xs:element name="term" type="xs:string"/><xs:element name="relationship" type="relationship" maxOccurs="unbounded"/></xs:sequence></xs:complexType><xs:complex Type name="entry"><xs:sequence><xs:element name="term" type="xs:string" minOccurs="0" maxOccurs="1"/><xs:element name="synonym" type="synonym" minOccurs="0" maxOccurs="unbounded"/></xs:sequence></xs:complexType><xs:element name="entry" type="entry"><xs:annotation><xs:documentation>
An entry in the thesaurus
</xs:documentation></xs:annotation></xs:element><xs:element name="thesaurus"><xs:annotation><xs:documentation>
Root of thesaurus.

</xs:documentation></xs:annotation><xs:complexType><xs:sequence minOccurs="0" maxOccurs="unbounded"><xs:element ref="entry"/></xs:sequence></xs:complexType></xs:element></xs:schema >

Annexe 4

Déclaration universelle des droits de l'Homme en français

Préambule

Considérant que la reconnaissance de la dignité inhérente à tous les membres de la famille humaine et de leurs droits égaux et inaliénables constitue le fondement de la liberté, de la justice et de la paix dans le monde.

Considérant que la méconnaissance et le mépris des droits de l'homme ont conduit à des actes de barbarie qui révoltent la conscience de l'humanité et que l'avènement d'un monde où les

êtres humains seront libres de parler et de croire, libérés de la terreur et de la misère, a été proclamé comme la plus haute aspiration de l'homme.

Considérant qu'il est essentiel que les droits de l'homme soient protégés par un régime de droit pour que l'homme ne soit pas contraint, en suprême recours, à la révolte contre la tyrannie et l'oppression.

Considérant qu'il est essentiel d'encourager le développement de relations amicales entre nations.

Considérant que dans la Charte les peuples des Nations Unies ont proclamé à nouveau leur foi dans les droits fondamentaux de l'homme, dans la dignité et la valeur de la personne humaine, dans l'égalité des droits des hommes et des femmes, et qu'ils se sont déclarés résolus à favoriser le progrès social et à instaurer de meilleures conditions de vie dans une liberté plus grande.

Considérant que les Etats Membres se sont engagés à assurer, en coopération avec l'Organisation des Nations Unies, le respect universel et effectif des droits de l'homme et des libertés fondamentales.

Considérant qu'une conception commune de ces droits et libertés est de la plus haute importance pour remplir pleinement cet engagement.

L'Assemblée générale proclame la présente Déclaration universelle des droits de l'homme comme l'idéal commun à atteindre par tous les peuples et toutes les nations afin que tous les individus et tous les organes de la société, ayant cette Déclaration constamment à l'esprit, s'efforcent, par l'enseignement et l'éducation, de développer le respect de ces droits et libertés et d'en assurer, par des mesures progressives d'ordre national et international, la reconnaissance et l'application universelles et effectives, tant parmi les populations des Etats Membres eux-mêmes que parmi celles des territoires placés sous leur juridiction.

Article premier

Tous les êtres humains naissent libres et égaux en dignité et en droits. Ils sont doués de raison et de conscience et doivent agir les uns envers les autres dans un esprit de fraternité.

Article 20

1. Toute personne a droit à la liberté de réunion et d'association pacifiques.
2. Nul ne peut être obligé de faire partie d'une association.

Article 21

1. Toute personne a le droit de prendre part à la direction des affaires publiques de son pays, soit directement, soit par l'intermédiaire de représentants librement choisis.
2. Toute personne a droit à accéder, dans des conditions d'égalité, aux fonctions publiques de son pays.
3. La volonté du peuple est le fondement de l'autorité des pouvoirs publics ; cette volonté doit s'exprimer par des élections honnêtes qui doivent avoir lieu périodiquement, au suffrage

universel égal et au vote secret ou suivant une procédure équivalente assurant la liberté du vote.

Article 22

Toute personne, en tant que membre de la société, a droit à la sécurité sociale ; elle est fondée à obtenir la satisfaction des droits économiques, sociaux et culturels indispensables à sa dignité et au libre développement de sa personnalité, grâce à l'effort national et à la coopération internationale, compte tenu de l'organisation et des ressources de chaque pays.

Article 23

1. Toute personne a droit au travail, au libre choix de son travail, à des conditions équitables et satisfaisantes de travail et à la protection contre le chômage.
2. Tous ont droit, sans aucune discrimination, à un salaire égal pour un travail égal.
3. Quiconque travaille a droit à une rémunération équitable et satisfaisante lui assurant ainsi qu'à sa famille une existence conforme à la dignité humaine et complétée, s'il y a lieu, par tous autres moyens de protection sociale.
4. Toute personne a le droit de fonder avec d'autres des syndicats et de s'affilier à des syndicats pour la défense de ses intérêts.

Article 24

Toute personne a droit au repos et aux loisirs et notamment à une limitation raisonnable de la durée du travail et à des congés payés périodiques.

Annexe 5

Extrait de la *Déclaration universelle des droits de l'Homme* en anglais

PREAMBLE

Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world,

Whereas disregard and contempt for human rights have resulted in barbarous acts which have outraged the conscience of mankind, and the advent of a world in which human beings shall enjoy freedom of speech and belief and freedom from fear and want has been proclaimed as the highest aspiration of the common people,

Whereas it is essential, if man is not to be compelled to have recourse, as a last resort, to rebellion against tyranny and oppression, that human rights should be protected by the rule of law,

Whereas it is essential to promote the development of friendly relations between nations,

Whereas the peoples of the United Nations have in the Charter reaffirmed their faith in fundamental human rights, in the dignity and worth of the human person and in the equal rights of men and women and have determined to promote social progress and better standards of life in larger freedom,

Whereas Member States have pledged themselves to achieve, in co-operation with the United Nations, the promotion of universal respect for and observance of human rights and fundamental freedoms,

Whereas a common understanding of these rights and freedoms is of the greatest importance for the full realization of this pledge,

Now, Therefore THE GENERAL ASSEMBLY proclaims THIS UNIVERSAL DECLARATION OF HUMAN RIGHTS as a common standard of achievement for all peoples and all nations, to the end that every individual and every organ of society, keeping this Declaration constantly in mind, shall strive by teaching and education to promote respect for these rights and freedoms and by progressive measures, national and international, to secure their universal and effective recognition and observance, both among the peoples of Member States themselves and among the peoples of territories under their jurisdiction.

Article 1.

- All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

Article 20.

- (1) Everyone has the right to freedom of peaceful assembly and association.
- (2) No one may be compelled to belong to an association.

Article 21.

- (1) Everyone has the right to take part in the government of his country, directly or through freely chosen representatives.
- (2) Everyone has the right of equal access to public service in his country.
- (3) The will of the people shall be the basis of the authority of government; this will shall be expressed in periodic and genuine elections which shall be by universal and equal suffrage and shall be held by secret vote or by equivalent free voting procedures.

Article 22.

- Everyone, as a member of society, has the right to social security and is entitled to realization, through national effort and international co-operation and in accordance

with the organization and resources of each State, of the economic, social and cultural rights indispensable for his dignity and the free development of his personality.

Article 23.

- (1) Everyone has the right to work, to free choice of employment, to just and favourable conditions of work and to protection against unemployment.
- (2) Everyone, without any discrimination, has the right to equal pay for equal work.
- (3) Everyone who works has the right to just and favourable remuneration ensuring for himself and his family an existence worthy of human dignity, and supplemented, if necessary, by other means of social protection.
- (4) Everyone has the right to form and to join trade unions for the protection of his interests.

Article 24.

- Everyone has the right to rest and leisure, including reasonable limitation of working hours and periodic holidays with pay.

Annexe 6

Extrait de la *Déclaration universelle des droits de l'Homme* en chinois

序言

鉴于对人类家庭所有成员的固有尊严及其平等的和不移的权利的承认，乃是世界自由、正义与和平的基础，

鉴于对人权的无视和侮蔑已发展为野蛮暴行，这些暴行玷污了人类的良心，而一个人人享有言论和信仰自由并免予恐惧和匮乏的世界的来临，已被宣布为普通人民的最高愿望，

鉴于为使人类不致迫不得已铤而走险对暴政和压迫进行反叛，有必要使人权受法治的保护，

鉴于有必要促进各国间友好关系的发展，

鉴于各联合国国家的人民已在联合国宪章中重申他们对基本人权、人格尊严和价值以及男女平等权利的信念，并决心促成较大自由中的社会进步和生活水平的改善，

鉴于各会员国业已誓愿同联合国合作以促进对人权和基本自由的普遍尊重和遵行，

鉴于对这些权利和自由的普遍了解对于这个誓愿的充分实现具有很大的重要性，

因此现在，

大会，

发布这一世界人权宣言，作为所有人民和所有国家努力实现的共同标准，以期每一个人和社会机构经常铭念本宣言，努力通过教诲和教育促进对权利和自由的尊重，并通过国家的和国际的渐进措施，使这些权利和自由在各会员国本身人民及在其管辖下领土的人民中得到普遍和有效的承认和遵行；

1 第一条

- 人人生而自由，在尊严和权利上一律平等。他们赋有理性和良心，并应以兄弟关系的精神相对待。

20 第二十条

- ㈠ 人人有权享有和平集会和结社的自由。
- ㈡ 任何人不得迫使隶属于某一团体。

21 第二十一条

- ㈠ 人人有直接或通过自由选择的代表参与治理本国的权利。
- ㈡ 人人有平等机会参加本国公务的权利。
- ㈢ 人民的意志是政府权力的基础；这一意志应以定期的和真正的选举予以表现，而选举应依据普遍和平等的投票权，并以不记名投票或相当的自由投票程序进行。

22 第二十二条

- 每个人，作为社会的一员，有权享受社会保障，并有权享受他的个人尊严和人格的自由发展所必需的经济、社会和文化方面各种权利的实现，这种实现是通过国家努力和国际合作并依照各国的组织和资源情况。

23 第二十三条

- ㈠ 人人有权工作、自由选择职业、享受公正和合适的工作条件并享受免于失业的保障。
- ㈡ 人人有同工同酬的权利，不受任何歧视。

- ㈢ 每一个工作的人，有权享受公正和合适的报酬，保证使他本人和家属有一个符合人的生活条件，必要时并辅以其他方式的社会保障。
- ㈣ 人人有为维护其利益而组织和参加工会的权利。

24 第二十四条

- 人人有享有休息和闲暇的权利，包括工作时间有合理限制和定期给薪休假的权利。

- 本宣言的任何条文，不得解释为默许任何国家、集团或个人有权进行任何旨在破坏本宣言所载的任何权利和自由的活动或行为。