Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés

François Daoust[1], Yves Marcoux[2]

[1]Informaticien au Centre ATO de l'Université du Québec à Montréal - Canada
[2]Professeur d'informatique à l'École de bibiothéconomie et des sciences de l'information, Université de Montréal - Canada

Résumé

.
Cet article présente une proposition de format d'échange de corpus à des fins de traitement par des logiciels de textométrie. Cette proposition, conforme aux recommandations du Text Encoding Initiative, a fait l'objet d'un accord de principe en août 2005 au sein du réseau ATONET. La proposition de base a déja permis de réaliser des passerelles de conversion des formats propriétaires de plusieurs logiciels. La proposition élargie, permettant le cumul des annotations, est susceptible d'orienter le développement futur des logiciels de textométrie.

Mots-clés : analyse de texte par ordinateur, lexicométrie, textométrie, normalisation des formats de documents, XML, TEI.

1. Contexte

Depuis plusieurs années, les chercheurs impliqués dans l'utilisation de l'ordinateur à des fins d'analyse textuelle se réunissent et collaborent en vue de faire connaître leurs outils, méthodes et pratiques d'analyse de texte assistée par ordinateur (ATO). Les communications scientifiques, notamment celles qui ont cours lors des *Journées internationales d'analyse des données textuelles* (JADT), permettent de saisir la nature complémentaire de plusieurs méthodes et programmes informatiques. Le temps est venu de se donner un cadre précis et concret pour évaluer la portée de ces méthodes et des logiciels qui les supportent, ce qui implique qu'on puisse facilement faire appel aux divers logiciels pour l'analyse d'un même corpus.

Pour développer des chaînes de traitement faisant appel à une variété de logiciels, il faut avoir la possibilité de transférer les données d'un logiciel à l'autre à l'autre sans perte des niveaux de description antérieures. Pour ce faire, il faut convenir de formats d'échange de documents électroniques en vue de leur traitement par les divers outils logiciels développés au sein de la communauté des chercheurs en ATO. L'utilisation du langage de balisage XML s'impose naturellement pour cette tâche. XML, rappelons-le, est un langage général de balisage des documents électroniques qui permet de publier, conserver, annoter et transformer des textes selon un protocole indépendant des formats propriétaires. La conversion des données, des logiciels et des interfaces à la norme XML facilite l'accès à l'ensemble de la chaîne de traitement textuelle : documentation et archivage sur la base d'une définition rigoureuse des données, ajout et maintenance de données provenant de diverses sources, interopérabilité des modules d'analyse, diffusion auprès de la communauté des chercheurs.

Cette question des formats de balisage des documents électroniques fait déjà l'objet de discussions dans plusieurs milieux académiques et industriels. C'est le cas en particulier au sein du «Text Encoding Initiative», (http://www.tei-c.org/), ce consortium formé par trois grandes associations scientifiques : l'*Association for Computers and the Humanities* (ACH), l'*Association for Computational Linguistics* (ACL) et l'*Association for Literary and*

Linguistic Computing (ALLC). On retrouve aussi ces discussions au sein de l'Organisation internationale de normalisation (ISO). C'est le cas, en particulier, du sous-comité connu sous le nom ISO/TC 37/SC4 (http://tc37sc4.org/). Il est impérieux que la communauté des chercheurs en ATO se positionne par rapport à ces propositions et convienne aussi de stratégies permettant le traitement de corpus balisés en XML par les divers logiciels d'analyse de texte par ordinateur. C'est là une condition pour que l'usage des technologies numériques en analyse de texte puisse devenir une activité courante.

C'est dans ce contexte qu'est née l'idée de regrouper les concepteurs de logiciels et de méthodes d'analyse afin de proposer un format commun pour l'échange de corpus annotés à des fins de traitement par des logiciels d'analyse de données textuelles. C'est ainsi qu'est né le réseau ATONET (http://www.atonet.net) appuyé par une subvention du gouvernement canadien pour le soutien de réseaux de chercheurs (Duchastel et al. 2004). Le réseau organise son travail autour de trois volets prioritaires de convergence technologique : un volet *méthodes et expérimentation*, un volet *normalisation XML des formats de documents électroniques* et un volet *terminologie*. Le premier volet concerne l'échange entre membres du réseau sur les méthodes et les ressources disponibles en ATO. Ces échanges impliquent qu'on convienne d'une banque de textes à des fins de test et de validation de nos méthodes et de nos logiciels. Le deuxième volet, dont nous rendons compte dans cette communication, concerne la définition de formats d'échange de documents électroniques en XML en vue de leur traitement par les divers outils logiciels développés au sein de la communauté des chercheurs en ATO. Le troisième volet est d'ordre terminologique et vise l'établissement d'un lexique de référence explicitant les termes du domaine en français et dans les autres langues des membres du réseau.

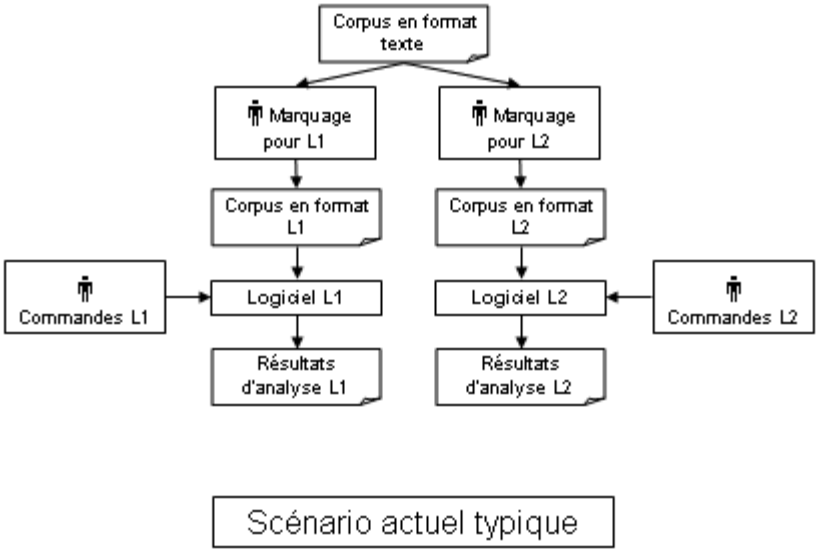
2. Interopérabilité de logiciels d'analyse de texte

Le problème de la normalisation des formats se pose sous plusieurs aspects.
- l'établissement d'un consensus autour de normes minimales de balisage XML constituant le format de référence pour l'échange des corpus à des fins de traitement par divers logiciels;
- l'écriture de passerelles permettant de passer des formats propriétaires au format XML et du format XML aux formats propriétaires afin de pouvoir utiliser les logiciels dans leur version actuelle;
- la discussion de stratégies permettant la manipulation de corpus ayant déjà fait l'objet de balisage XML suivant des DTD (description du type de documents) définies par d'autres groupes d'intérêt;
- la discussion de formats permettant l'exportation ou l'importation des résultats produits par les logiciels d'analyse textuelle : marquage de segments textuels, production de tableaux lexicaux, graphes et données lexicales.

Ces différents aspects, et l'apport potentiel des propositions que nous formulons dans cette communication, sont mis en évidence dans différents scénarios d'interopérabilité de logiciels d'analyse de texte. Trois scénarios sont présentés ci-dessous. Le premier correspond à la situation actuelle, avec des logiciels possédant chacun son format propriétaire. Le deuxième illustre l'interopérabilité rendue possible par le format correspondant à notre *proposition de base*. Le troisième illustre les possibilités visées par le format correspondant à notre *proposition élargie*.

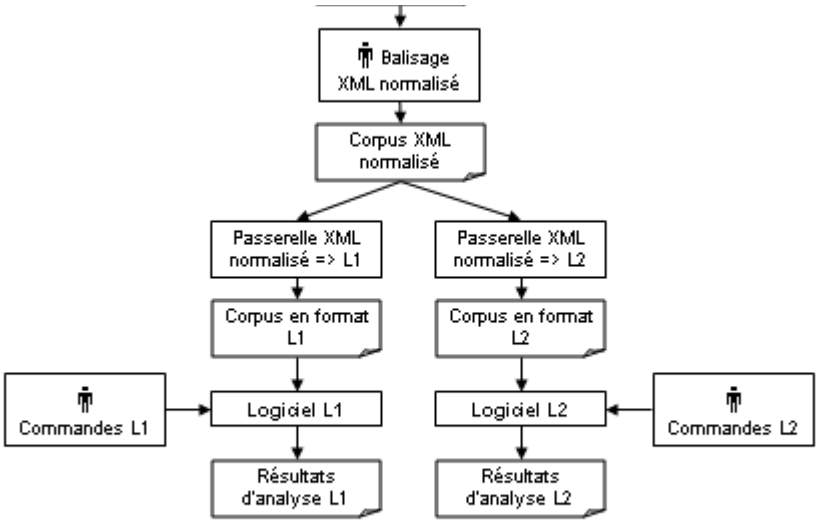
Chaque scénario présenté fait intervenir deux logiciels seulement, mais ils suffisent à illustrer la situation où le nombre de logiciels est arbitraire. Les étapes demandant une intervention humaine sont indiquées par une icône de personne.

Dans le premier scénario, l'interopérabilité est inexistante. Pour analyser un même corpus avec deux logiciels différents, il faut procéder à deux préparations différentes pour rendre le corpus conforme aux deux formats propriétaires. Il n'y a pas de possibilité d'analyses en « cascade », pour lesquelles le résultat d'une analyse faite avec un logiciel servirait d'intrant à une deuxième analyse avec un autre logiciel.



Scénario actuel typique

Dans le deuxième scénario, on a introduit un format « pivot » neutre, normalisé, basé sur XML. Au lieu de devoir préparer le corpus pour chaque logiciel, une seule préparation est requise. Cependant, les analyses en cascade sont toujours impossibles. Notre *proposition de base* (cf. section 4) vise à fournir un tel format pivot. Nous avons indiqué une préparation manuelle du XML normalisé, mais en pratique, ce format peut être obtenu par conversion automatique d'un format propriétaire.

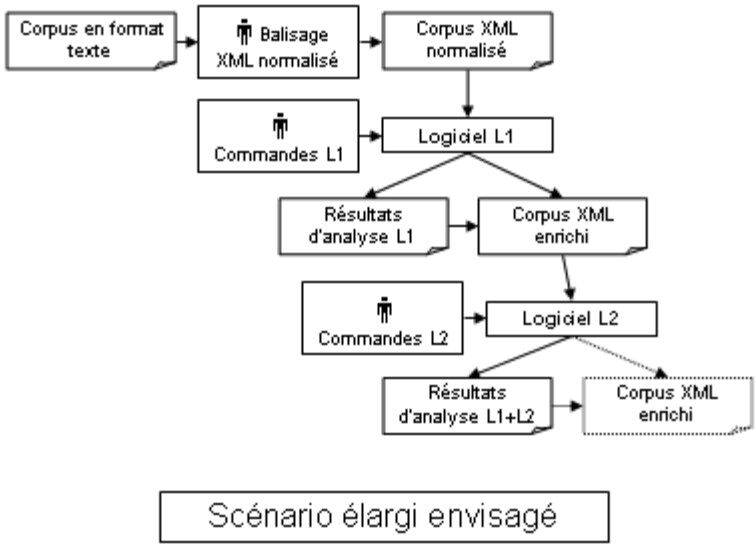


Scénario de base envisagé

Notons que ce scénario ne présuppose ni n'exige aucune modification aux logiciels individuels; il nécessite seulement le développement de passerelles de conversion du format pivot vers chacun des formats propriétaires.

Le développement de passerelles de conversion *dans la direction opposée*, c'est-à-dire d'un format propriétaire vers le format pivot, est aussi intéressant, dans la mesure où beaucoup de corpus actuellement disponibles le sont dans un format propriétaire. Convertir ces corpus vers le format pivot les rend instantanément traitables par les autres logiciels.

Le troisième scénario illustre une véritable interopérabilité des logiciels. Il implique cependant que les logiciels soient modifiés pour tenir compte d'éventuels traitements antérieurs, par exemple le découpage en mots effectué par un autre logiciel. Les logiciels devront aussi pouvoir intégrer en tout ou en partie au corpus originel les résultats de leur propres traitements, résultant en un *corpus XML enrichi*. Cette intégration peut être plus ou moins serrée. À une extrémité du spectre des possibilités, l'intégration peut se résumer à l'insertion dans l'entête du corpus d'un pointeur vers un fichier externe contenant les résultats. À l'autre extrémité, l'intégration peut se faire par l'insertion de nouvelles balises dans le corpus initial ou dans des fichiers d'annotation associés. Notre *proposition élargie* (cf. section 5) vise essentiellement à fournir un format adéquat de corpus XML enrichi.



Scénario élargi envisagé

3. Des formats propriétaires au balisage XML-TEI

La première tâche du groupe de travail sur les formats d'échange des corpus a été de faire l'inventaire des formats actuellement utilisés par les divers logiciels des membres du réseau ATONET. Il s'agit des logiciels ALCESTE (Reinert), Astartex (Viprey) DTM (Lebart), LEXICO3 (Salem et al.), SATO (Daoust) et Weblex (Heiden). Dans une première phase d'analyse, nous nous sommes concentrés sur les logiciels *pré-XML* offrant des versions distribuées : ALCESTE, DTM, LEXICO et SATO.

Ces divers logiciels utilisent des *formats propriétaires*, c'est-à-dire des formats qui ne sont utilisés que par le logiciel. Si la syntaxe utilisée varie d'un logiciel à l'autre, on retrouve une certaine équivalence logique entre les diverses fonctions de marquage. En particulier, tous ces logiciels procèdent à un découpage en occurrences (*token*) de formes lexicales (*type*) dont la fréquence dans différentes parties du texte servira de base aux algorithmes de lexicométrie. Donc, on retrouve deux types de segmentation du corpus : découpage en *token* et partitionnement du corpus sensible à des marques paratextuelles qui balisent l'organisation du corpus en documents, locuteurs et profils sociologiques des segments rassemblés dans le corpus. Selon les logiciels, ces *variables externes*, sont appelées *clés*, *propriétés*, *variables exogènes* ou *réponses à des questions fermées*. Selon les logiciels aussi, ces variables sont typées ou non typées, imbriquées dans le corpus ou inscrites dans un fichier de données externe.

Le découpage en tokens suit aussi des règles diverses. Il est généralement paramétrable et fait l'objet de traitements différents selon les logiciels concernant les problèmes classiques de l'ambiguïté orthographique : apostrophes, ponctuations, majuscules, traits d'union, fins de ligne, locutions, lemmatisation et méta-caractères. Certains logiciels offrent des ressources de type dictionnaires, locutions, filtrage, analyse morphologique, etc. À ce niveau donc, on fait face à une grande diversité et à un ensemble de contraintes spécifiques. L'objectif d'un

format d'échange n'est pas d'imposer une solution unique à ces problèmes, mais d'offrir une façon unique de noter les résultats du traitement.

La stratégie adoptée consiste à convenir d'un format pivot basé sur des formats de balisage normalisés et de pratiques de balisage ayant fait l'objet de consultations publiques. Pour utiliser ce format d'exportation unique, il n'est pas nécessaire d'exiger qu'il soit adopté de façon native par chacun des développeurs de logiciels. En fait, il est possible de concevoir des programmes externes de conversion agissant comme passerelles entre chacun des formats propriétaires et le format pivot. On a donc besoin de deux types de passerelles : du format propriétaire au format pivot, et du format pivot vers le format propriétaire. Cette solution, immédiatement applicable, ne pourra cependant aller au-delà des contraintes actuelles des logiciels. Voilà pourquoi nous proposons un format minimal, conforme à ces contraintes, et un format optimal, orientant le développement futur de chaque logiciel de telle sorte qu'il puisse respecter les annotations existantes découlant de décisions antérieures.

Pour ce qui est du formalisme général du format pivot, nous nous sommes tournés, tout naturellement, vers la norme XML qui fait l'objet d'un large consensus et pour laquelle on trouve de plus en plus d'outils de traitement informatique. Rappelons les principes généraux à la base de l'*Extensible Markup Language* (XML).

1. Un texte est un objet composite dont les composantes doivent être distinguées;
2. On sépare le contenu logique du document de sa présentation visuelle;
3. On utilise un balisage (marquage) symbolique qui suit une syntaxe générale simple, facile à décoder par un programme et totalement explicite pour un lecteur humain;
4. On distingue entre *document bien formé,* soumis aux contraintes minimales du langage XML, et *document valide* soumis aux contraintes supplémentaires définissant un type particulier de document;
5. XML est donc un formalisme général extensible permettant de définir des types particuliers de documents électroniques à des fins d'échange au sein de communautés particulières;
6. Les données appartiennent aux utilisateurs (communauté d'intérêts) qui sont responsables de définir leurs types de document en utilisant des formalismes syntaxiques : DTD (définition du type de document) ou autres.

Les contraintes syntaxiques minimales du document XML bien formé sont les suivantes.

1. Le codage des caractères repose sur la norme UNICODE, mais peut faire appel à diverses formes d'encodage spécifiées en entête. Les caractères réservés sont < , >. Ces caractères délimitent les balises qui structurent le contenu du document. Aussi des suites de caractères sans espaces débutant par *&* et se terminant par *;* seront interprétées comme des *appels d'entités* permettant d'encoder tout caractère UNICODE ou d'inclure divers objets préalablement définis : chaînes de caractères, fichiers, etc.
2. Tout élément de contenu doit être délimité à gauche par une balise ouvrante (<nom-de-balise>) et à droite par une balise fermante (</nom-de-balise>); on peut avoir une contraction des deux s'il n'y a pas de contenu entre la balise ouvrante et la balise fermante : on parle alors de *balise d'élément vide* ou *auto-fermante* (<nom-de-balise/>);
3. Tout document doit être contenu dans un élément racine qui englobe l'ensemble du document;
4. La structure d'un document est strictement hiérarchique : tout élément à l'intérieur d'un élément supérieur doit être totalement inclus dans l'élément supérieur;

5. Une balise peut porter des attributs qui en précisent l'interprétation.

La contrainte qui impose que le balisage soit strictement hiérarchique ne pose pas problème quand il s'agit de représenter la structure formelle d'un document : collection, livre, chapitre, section, paragraphe, etc. La situation est très différente lorsqu'il s'agit de représenter des structures analytiques. Par exemple, s'il est naturel de représenter la structure formelle d'un poème en termes de stophes et lignes, on voudra aussi analyser le poème en termes de phrases, propositions, syntagmes, etc. Or, cette structure linguistique est indépendante de la structure formelle. Ainsi, la phrase peut chevaucher la frontière des lignes de même qu'elle peut ne contenir que des parties de la ligne. Plusieurs solutions peuvent être envisagées pour contourner ce problème, notamment l'utilisation de balises vides qui marquent des repères en laissant à l'application le soin d'en interpréter la portée. Il est aussi possible de définir des annotations qui construisent des structures analytiques en référant à des balises structurelles ou à des balises vides.

Le *Text Encoding Initiative* distingue trois types de structures qui rendent compte de l'idée de corpus en tant qu'objet d'analyse.

1. La structure formelle qui correspond à la division en documents, sections, chapitres, paragraphes, etc. ;

2. L'information descriptive sur le contexte de production du corpus; c'est l'entête TEI qui contient les références bibliographiques, les décisions d'encodage, l'histoire des traitements, etc. ;

3. L'annotation linguistique : catégories grammaticales, relations de cohésion, et annotation analytique en général.

Le TEI travaille depuis longtemps sur le marquage des corpus. Notre groupe de travail a donc voulu vérifier s'il était possible de s'appuyer sur les recommandations du TEI pour proposer un noyau minimal de règles de balisage XML des corpus permettant de rendre compte du codage propriétaire utilisé par chacun des logiciels. Rappelons d'abord que le TEI ne propose pas un seul modèle de document, mais un ensemble de façons de faire alternatives. Parmi ces façons de faire, nous avons privilégié les formes de balisage les plus génériques qui correspondent aussi à la sémantique des logiciels qui considèrent les corpus comme un objet formel donnant lieu à un tableau de données croisant des *variables* (fréquences lexicales, variables catégorielles ou numériques) et des individus statistiques : documents, segments ou parties de texte.

4. Proposition de base

La proposition minimale de balisage XML-TEI ne vise pas à aller au-delà du commun dénominateur permettant de soumettre un corpus à chacun des logiciels considérés. Les avis produits par les logiciels sous forme de rapports serviront donc directement à l'interprétation par l'analyste sans mécanisme spécifique de réinjection des résultats enrichissant le balisage original.

De la structure formelle du document, seuls deux éléments seront interprétés et traduits : <**p**>...</**p**> qui marque les paragraphes et <**c**>...</**c**>qui sera utilisé pour transmettre des caractères sans conversion de casse.

De l'entête TEI, seule la balise <title>...</title> sera interprétée pour identifier le corpus dans le format propriétaire, lorsque requis.

Les autres balises interprétées par la passerelle appartiennent à la catégorie des *milestone* dans la terminologie TEI (Proposition 5 , section 6.10.3 *Milestone Tags*). Voici la signification de ces balises :

- **<milestone/>** indique une nouvelle section de texte à l'intérieur d'un certain système référentiel nommé par l'attribut *unit;*
- **<pb/>** indique le début d'une nouvelle page;
- **<lb/>** indique le début d'une nouvelle ligne;
- **<cb/>** indique la frontière entre deux colonnes de texte; ces colonnes permettront notamment de séparer les réponses ouvertes d'un individu pour le logiciel DTM.

En principe, tout document TEI est exportable vers les formats propriétaires. Cependant, seules certaines balises sont impliquées dans la conversion, tout le reste étant ignoré. Par conséquent, un document TEI qui utiliserait d'autres balises pour représenter une segmentation admissible aux traitements par les logiciels considérés ici devrait d'abord être soumis à un mécanisme de transformation (feuille XSLT ou autre) qui ajoutera les *milestone* pertinents. Comme les *milestone,* en tant que balises vides, n'interfèrent pas avec le balisage hiérarchique, on peut très bien les faire coexister avec le balisage original.

Examinons la proposition à partir d'un exemple de codage du poème *Le dormeur du val* d'Arthur Rimbaud.

Exemple de texte en XML-TEI avec balises <milestone>

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!DOCTYPE TEI SYSTEM "..\dtd\tei.dtd" [
<!ENTITY % TEI.header "INCLUDE"> <!ENTITY % TEI.core "INCLUDE"> <!ENTITY %
TEI.textstructure "INCLUDE"> <!ENTITY % TEI.analysis "INCLUDE"> <!ENTITY % TEI.iso-fs
"INCLUDE"> <!ENTITY % TEI.linking "INCLUDE"> ]>
```

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
```

```
<teiHeader>
<fileDesc>
<titleStmt><title>Le dormeur du val</title><author>Arthur Rimbaud</author></titleStmt>
<publicationStmt> <p>Publié par...</p></publicationStmt>
<sourceDesc> <p>Texte fourni par ... </p></sourceDesc>
</fileDesc>

<encodingDesc>
<refsDecl>
<p>Les balises «milestone n="valeur-de variable" unit="nom-de-variable"» concernent les mots qui
suivent la balise jusqu'à l'apparition d'un nouveau milestone de même unit. Les références de pagination
utilisent les balises pb (début de page) et lb (début de ligne) et cb (frontière de colonne).</p>

<p>milestone champ "titre" "poème" "signature"</p>
</refsDecl>
</encodingDesc>
</teiHeader>
```

```
<text>
<body>
<pb n="rimbaud-le_dormeur_du_val/1"/>

<p><lb n="1"/><milestone n="titre" unit="champ"/> Le dormeur du val </p>

<p><lb n="2"/><milestone n="poème" unit="champ"/> C'est un trou de verdure où chante une rivière
<lb n="3"/>Accrochant follement aux herbes des haillons
<lb n="4"/>D'argent ; où le soleil, de la montagne fière,
<lb n="5"/>Luit : c'est un petit val qui mousse de rayons.</p>

<p><lb n="6"/>Un soldat jeune, bouche ouverte, tête nue,
<lb n="7"/>Et la nuque baignant dans le frais cresson bleu,
<lb n="8"/>Dort ; il est étendu dans l'herbe, sous la nue,
<lb n="9"/>Pâle dans son lit vert où la lumière pleut.</p>

<p><lb n="10"/>Les pieds dans les glaïeuls, il dort. Souriant comme
<lb n="11"/>Sourirait un enfant malade, il fait un somme :
<lb n="12"/>Nature, berce-le chaudement : il a froid. </p>

<p><lb n="13"/>Les parfums ne font pas frissonner sa narine ;
<lb n="14"/>Il dort dans le soleil, la main sur sa poitrine
<lb n="15"/>Tranquille. Il a deux trous rouges au côté droit. </p>

<p><lb n="16"/><milestone n="signature" unit="champ"/>Arthur Rimbaud</p>
</body>
</text>
```

```
</TEI>
```

Dans cet exemple, on a mis dans des cases séparées diverses composantes du document en format TEI. Examinons-les une à une.

1. *<?xml* ... C'est le prologue recommandé pour indiquer que le document est en XML; on y indique ici que le document utilise un encodage des caractères iso-latin-1 (iso-8859-1).
2. *<!DOCTYPE TEI* ... C'est la définition du type de document indiquant ici qu'il s'agit d'un document TEI. Dans cet exemple, on fait référence à une DTD qui se trouve sur le disque local. C'est pour des fins de validation : on peut lancer un parseur XML qui validera le document et indiquera les erreurs éventuelles. En production, on ne contente de référer à une déclaration publique de la DTD standard du TEI. Il est aussi à noter qu'il existe un type de document *teiCorpus* qui permet de rassembler un ensemble de textes possédant chacun un entête TEI.
3. *<TEI* ... Voilà l'élément racine qui porte le nom du type de document annoncé dans le DOCTYPE. L'attribut *xmlns* signifie que les éléments qui suivent sont, sauf mention contraire, définis dans un espace de noms portant la signature du TEI. Cela permet de ne pas confondre ces noms avec des homographes qui pourraient appartenir à d'autres DTD.
4. *<teiHeader>* ... C'est l'entête TEI, lui-même composé de plusieurs sections.
   La section <fileDesc> contient une sous-section <titleStmt> qui fournit le titre et l'auteur du texte, une sous-section <publicationStmt> et une sous-section <sourceDesc>.
   La section <encodingDesc> permet de documenter la codification du texte. En particulier, on y retrouve la sous-section <refsDecl> qui décrit les systèmes

référentiels, notamment les balises *milestone* utilisées pour introduire les variables qui traduisent les balises LEXICO, les propriétés SATO, etc.
Plusieurs sections facultatives de l'entête TEI ne sont pas illustrées dans cet exemple. Elles permetttent de conserver un ensemble de métadonnées de description et de classification.

5. <*text*> ... C'est ici que débute le texte proprement dit. Dans la section <*body*>, qui nous intéresse ici plus particulièrement, on peut retrouver, outre le texte brut, un nombre important de balises TEI dont certaines seulement seront interprétées par nos logiciels non-XML. Les autres balises sont supprimées par les passerelles de traduction, ou sont considérées comme des commentaires. Voici le descriptif des balises utilisées dans l'exemple.
    - La référence de pagination (nom du document ou/et numéro de page) est transmise comme valeur de l'attribut *n* de la balise vide de début de page *pb*;
    - Les débuts de ligne sont marquées par la balise vide *lb*. On peut utiliser l'attribut *n* de la balise pour indiquer le numéro de la ligne;
    - Les paragraphes sont marqués par les balises <*p*> </*p*>;
    - Les variables sont représentées par des balises vides *milestone*. L'attribut *unit* est utilisé pour indiquer le nom de la variable, *champ* dans notre exemple, dont la valeur est transmise par l'attribut *n*.
6. </*TEI*> ... C'est la balise de fermeture du document TEI.

L'entête TEI est un ajout important par rapport aux formats propriétaires utilisés par les logiciels considérés. C'est un élément obligatoire de tout document TEI. La conversion du format propriétaire vers le format TEI doit se contenter de générer un entête minimal à compléter manuellement par la suite.

5. Proposition élargie

Les logiciels considérés dans notre analyse procèdent tous à un découpage en mots et à la constitution d'un inventaire des formes lexicales avec décomptes des fréquences et avec des propriétés statistiques et catégorielles. Aussi, divers analyseurs délimitent des segments sur le texte correspondant à des caractéristiques statistiques. Dans la proposition XML-de base, rien ne permet cependant de rendre compte du résultat de ces analyses en termes d'annotations sur le corpus. Voilà pourquoi il nous est apparu nécessaire de proposer un balisage XM-TEI qui permette de conserver des annotations lexicales et contextuelle découlant de l'analyse. Ainsi, dans le développement futur de leurs outils, les concepteurs de logiciels pourront choisir de tenir compte de ces nouvelles possibilités.

Puisque le découpage en mots (*token*) est l'opération initiale des logiciels procédant à partir du texte brut, il est nécessaire de pouvoir référer à ce découpage de façon non ambigu. Il est aussi souhaitable que les logiciels puissent respecter un découpage préalable, s'il existe, afin de permette de disposer d'annotations compatibles. Les recommandations du TEI prévoient déjà une balise permettant d'identifier ce qu'est un *mot.* C'est la balise <**w**>..</**w**>, L'attribut *id*, ou sa généralisation XML *xml:id* permet d'associer au *token* un identificateur unique permettant de référer sans ambiguïté à chaque occurrence.

Il est important de noter la distinction entre *token* et *word form* telle que précisée dans la proposition de l'ISO/TC37/SC4 sur l'annotation morpho-syntaxique. Le *token* y est défini comme une séquence discursive continue et non-vide qui résulte d'une segmentation du texte par repérage de séparateurs ou morphèmes. Le *word form* est plutôt défini comme une unité linguistique référant à des *token* qui peuvent être discontinus. Un *word form* peut correspondre à une position vide (zéro token) et un *token* peut faire partie de plusieurs *word*

form. Un logiciel de textométrie pourrait donc permettre d'utiliser le *word form* plutôt que le *token* comme unité lexicométrique. Mais, il reste qu'à la base, on doit disposer d'un système référentiel sachant qu'il ne règle pas à lui seul le problème des unités linguistiques. Le choix du niveau de granularité pour le découpage en *token* n'appartient pas à la norme.

Voici un extrait du poème *Le dormeur du val* avec balisage TEI des *token*.

Exemple de texte en XML-TEI avec balises <milestone> et <w>

```
<text>
<body>
<pb n="rimbaud-le_dormeur_du_val/1"/>

<p><lb n="1"/><milestone n="titre" unit="champ"/> <w xml:id="w2" n="1">Le</w> <w xml:id="w3"
n="2">dormeur</w> <w xml:id="w4" n="3">du</w> <w xml:id="w5" n="4">val</w> </p>

<p><lb n="2"/><milestone n="poème" unit="champ"/> <w xml:id="w7" n="1">C'</w><w xml:id="w8"
n="2">est</w> <w xml:id="w9" n="3">un</w> <w xml:id="w10" n="4">trou</w> <w xml:id="w11"
n="5">de</w> <w xml:id="w12" n="6">verdure</w> <w xml:id="w13" n="7">où</w> <w xml:id="w14"
n="8">chante</w> <w xml:id="w15" n="9">une</w> <w xml:id="w16" n="10">rivière</w>
<lb n="3"/><w xml:id="w18" n="1">Accrochant</w> <w xml:id="w19" n="2">follement</w> <w
xml:id="w20" n="3">aux</w> <w xml:id="w21" n="4">herbes</w> <w xml:id="w22" n="5">des</w> <w
xml:id="w23" n="6">haillons</w>
<lb n="4"/><w xml:id="w25" n="1">D'</w><w xml:id="w26" n="2">argent</w> <w xml:id="w27"
n="3">;</w> <w xml:id="w28" n="4">où</w> <w xml:id="w29" n="5">le</w> <w xml:id="w30"
n="6">soleil</w><w xml:id="w31" n="7">,</w> <w xml:id="w32" n="8">de</w> <w xml:id="w33"
n="9">la</w> <w xml:id="w34" n="10">montagne</w> <w xml:id="w35" n="11">fière</w><w
xml:id="w36" n="12">,</w>
<lb n="5"/><w xml:id="w38" n="1">Luit</w> <w xml:id="w39" n="2">:</w> <w xml:id="w40"
n="3">c'</w><w xml:id="w41" n="4">est</w> <w xml:id="w42" n="5">un</w> <w xml:id="w43"
n="6">petit</w> <w xml:id="w44" n="7">val</w> <w xml:id="w45" n="8">qui</w> <w xml:id="w46"
n="9">mousse</w> <w xml:id="w47" n="10">de</w> <w xml:id="w48" n="11">rayons</w><w
xml:id="w49" n="12">.</w> </p>
<body>
</text>
```

Le découpage en mots est marqué par la paire de balises <***w***> </***w***>. Chaque mot possède un identificateur unique, ***xml:id****="w2" par exemple,* qui permet de le référencer. On peut non seulement pointer de façon unique chacun des *token*, mais on on peut aussi utiliser des balises, telles la balise TEI *span*, pour désigner des segments de texte : par exemple,
**
désigne le syntagme *la montagne fière*. Comme on a donné un identificateur unique au *span*, on pourra également y faire référence.

Les logiciels qui procèdent à des analyses lexicométriques doivent nécessairement découper le texte en *token*. Dans cette représentation, on ne fait que rendre explicite ce découpage. Il serait souhaitable que ce découpage, s'il est déjà marqué dans le corpus, puisse être respecté par les logiciels de traitement lexical afin de produire des annotations compatibles.Une fois qu'on dispose d'un système référentiel explicite, il devient donc facile de gérer des structures d'annotations renvoyant aux identificateurs de *token*. Cette annotation peut faire partie du corpus ou elle peut faire partie de documents d'annotations indépendants utilisant le système référenciel du corpus. En particulier, les structures de traits, faisant l'objet d'une recommndation conjointe ISO-TEI, permettent de gérer simplement les propriétés atomiques des *token* et des formes lexicales. Voici un exemple de fichier externe qui annote les premiers mots du *dormeur du val* et les formes lexicales correspondantes.

Exemple de fichier externe d'annotations

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE TEI SYSTEM "..\dtd\tei.dtd" [
<!ENTITY % TEI.header "INCLUDE"> <!ENTITY % TEI.core "INCLUDE"> <!ENTITY %
TEI.textstructure "INCLUDE"> <!ENTITY % TEI.analysis "INCLUDE"> <!ENTITY % TEI.iso-fs
"INCLUDE"> <!ENTITY % TEI.linking "INCLUDE"> ]>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt><title>Le dormeur du val</title><author>Arthur Rimbaud </author></titleStmt>
<publicationStmt> <p>Publié par...</p></publicationStmt>
<sourceDesc> <p>Texte fourni par ... </p></sourceDesc>
</fileDesc>
<encodingDesc>
<fsdDecl type="prolex" url="poeme_fsd.xml"/>
<fsdDecl type="protex" url="poeme_fsd.xml"/>
</encodingDesc>
</teiHeader>
<text>
<body>
<linkGrp type="prop" targFunc="token prolex protex">
<link targets="poemes.xml#w2 #px74 #pw2">
<link targets="poemes.xml#w3 #px41 #pw3">
<link targets="poemes.xml#w4 #px45 #pw4">
<link targets="poemes.xml#w5 #px147 #pw5">
<!-- etc... -->
</linkGrp>
<p>
<fs xml:id="px74 type="prolex" n="le"><f name="Fréqtot"><numeric value="25"/></f></fs>
<fs xml:id="px41" type="prolex" n="dormeur"><f name="Fréqtot"><numeric
value="1"/></f></fs>
<fs xml:id="px45" type="prolex" n="du"><f name="Fréqtot"><numeric value="9"/></f></fs>
<fs xml:id="px147" type="prolex" n="val"><f name="Fréqtot"><numeric value="3"/></f></fs>

<fs xml:id="pw2" type="protex"><f name="Gramr"><symbol value="Artdéf"/></f></fs>
<fs xml:id="pw3" type="protex"><f name="Gramr"><symbol value="Nomcom"/></f></fs>
<fs xml:id="pw4" type="protex"><f name="Gramr"><symbol value="Artdéf"/></f></fs>
<fs xml:id="pw5" type="protex"><f name="Gramr"><symbol value="Nomcom"/></f></fs>
</p>
</body>
</text>
</TEI>
```

Examinons les composantes de ce fichier.

1. *<?xml* ... C'est le prologue habituel.
2. *<!DOCTYPE TEI* ... C'est la définition du type de document indiquant ici qu'il s'agit d'un document TEI avec les modules *iso-fs* et *linking* pour les liens.

3. <*TEI* ... C'est l'élément racine.
4. <*teiHeader*> ... C'est l'entête TEI. Les définitions de *milestone* sont disparues au profit de définitions de structures de traits contenues dans des fichiers externes. Par exemple, <*fsdDecl type="prolex" url="poeme_fsd.xml"/*> indique que la définition se trouve dans le fichier *poeme_fsd.xml*.
5. <*body*> ... C'est ici que se trouve les annotations. On a d'abord un <*linkGrp type="prop" targFunc="token prolex protex"*> qui contient une suite de <*link targets="poemes.xml#w3 #px41 #pw3"*> qui relient un mot (<*w*>) à une structure de traits lexicaux et à une structure de traits contextuels. Par exemple, on lie le mot *w3* contenu dans le fichier *poemes.xml*, la structure de traits lexicaux *px41* et la structure de traits contextuels *pw3*.
   Examinons une structure de traits. La structure est introduite par la balise <*fs xml:id="px41" type="prolex" n="dormeur"*>. Dans cet exemple, on indique qu'il s'agit de traits lexicaux (*prolex*). L'attribut *n* est utilisé ici pour indiquer la forme lexicale normalisée au niveau de la casse. On a ensuite les traits eux-mêmes. Par exemple <*f name="Fréqtot"*><*numeric value="2"/*><*/f*> contient la fréquence totale du lexème; <*f name="Gramr"*><*symbol value="Nomcom"/*> indique que l'occurrence porte la catégorie grammaticale *Nomcom* (nom commun).
6. <*/TEI*> ... C'est la balise de fermeture du document TEI.

Voici, finalement le contenu du fichier *poeme_fsd.xml* qui tient lieu de dictionnaire de données pour les structures de traits. L'existence de ce fichier n'est pas obligatoire, mais est fortement recommandée par le TEI. Si un tel dictionnaire est utile du point de vue documentaire, il est aussi très précieux pour le programme qui doit lire le fichier d'annotations et qui saura en partant à quoi il doit s'attendre...

Exemple de fichier de déclarations de systèmes de traits

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!DOCTYPE teiFsd2 SYSTEM "..\dtd\declarefs.dtd" [
<!ENTITY % TEI.XML "INCLUDE">] >
```

```
<teifsd>
```

```
<teiHeader>
<fileDesc>
<titleStmt><title>Le dormeur du val</title><author>Arthur Rimbaud </author></titleStmt>
<publicationStmt> <p>Publié par...</p></publicationStmt>
<sourceDesc> <p>Texte fourni par ... </p></sourceDesc>
</fileDesc>
</teiHeader>
```

```
<fsDecl type="prolex">
<fsDescr>Définition des propriétés lexicales</fsDescr>
<fDecl name="Fréqtot" org="unit"/>
<fDescr>Fréquence totale pour l'ensemble du corpus</fDescr>
<vRange><numeric value="0" max="65535" trunc="true"/></vRange>
<vDefault><numeric value="0"/></vDefault>
</fDecl>
</fsDecl>

<fsDecl type="protex">
<fsDescr>Définition des propriétés textuelles</fsDescr>
<fDecl name="Gramr" org="set"/>
<fDescr>Catégorie grammaticale en contexte</fDescr>
<vRange><symbol value="nil"/> <symbol value="Abr"/> <symbol value="Adjdém"/>
<symbol value="Adjexc"/> <symbol value="Adjind"/> <symbol value="Adjint"/>
<symbol value="Adjnum"/> <symbol value="Adjpos"/><symbol value="Adjqua"/>
<symbol value="Adjrel"/><symbol value="Adv"/><symbol value="Artdéf"/>
<symbol value="Artind"/><symbol value="Artpar"/><symbol value="Con"/>
<symbol value="Dél"/><symbol value="Int"/><symbol value="Mor"/>
<symbol value="Nomcom"/><symbol value="Nompro"/><symbol value="Ono"/>
<symbol value="Pon"/><symbol value="Pré"/><symbol value="Prodém"/>
<symbol value="Proexc"/><symbol value="Proind"/><symbol value="Proint"/>
<symbol value="Proper"/><symbol value="Propos"/><symbol value="Proréf"/>
<symbol value="Prorel"/><symbol value="Rés"/><symbol value="X"/>
<symbol value="Vaux"/><symbol value="Vconj"/><symbol value="Vinf"/>
<symbol value="Vparpas"/><symbol value="Vparpré"/></vRange>
<vDefault><symbol value="nil"/></vDefault>
</fDecl>
</fsDecl>
```

```
</teifsd>
```

Dans sa forme générale, ce fichier ressemble à une suite de structures de traits qui contiendrait toutes les valeurs admissibles de chacun des traits. Examinons les composantes du fichier.

1. *<?xml* ... C'est le prologue habituel.
2. *<!DOCTYPE teifsd* ... C'est la définition du type de document indiquant ici qu'il s'agit d'un document *teifsd*.
3. *<teifsd* ... C'est l'élément racine.
4. *<teiHeader>* ... C'est l'entête TEI dans une forme simplifiée.
5. *<fsDecl* ... Introduit les déclarations de structures de traits, une pour chacun des deux types : *prolex* et *protex*. On a ensuite la définition de chaque trait. La balise *<vRange>* introduit les valeurs posssibles tandis que *<vDefault>* permet de définir la valeur par défaut.
6. *</teifsd>* ... C'est la balise de fermeture du document teifsd.

On aura remarqué que les fichiers XML sont très *verbeux*. Les structures, très redondantes, sont aussi très régulières, ce qui simplifie le traitement informatique. Comme les librairies de *parsage* sont disponibles dans les langages usuels, on peut y faire systématiquement appel en raison même de la régularité de la syntaxe. Au niveau du stockage des fichiers, les algorithmes de compression de type *zip* et autres permettent des niveaux de compression très élevés, ce qui permet de réduire la taille des fichiers à des fins de stockage et d'échange.

6- Conclusion et travaux à venir

La proposition de normalisation XML-TEI du format d'échange des corpus à des fins de traitement est déjà pleinement fonctionnelle grâce aux passerelles de conversion. Ces passerelles ont d'ailleurs fait l'objet d'une première expérimentation qui a donné lieu à une contribution aux JADT 2006 (cf. Gélinas-Chebat, Daoust, Dufresne, Dobrowolski, Gallopel). La proposition élargie, qui implique d'adapter les logiciels existants, est encore à implanter. On veut l'expérimenter pour développer des liens directs entre logiciels aux niveaux des fonctionnalités elle-mêmes : par exemple, un calcul de spécificités commandé directement à LEXICO depuis l'analyseur *distance* de SATO. Aussi, une des priorités de la deuxième année du réseau ATONET sera de développer des modules pour la gestion des métadonnées et de l'entête TEI. Il y a l'entête initial qui définit les origines du corpus et ses règles de codification. Il y a aussi l'entête évolutif qui permet de documenter la valeur ajoutée par les traitements au fur et à mesure de leur application. Il reste donc encore bien des défis a relever. Cependant, on peut considérer que ce premier effort de normalisation constitue un acquis très important qui montre qu'on peut avancer dans les protocoles d'échange sans attendre d'avoir résolu tous les problèmes.

Références

Daoust, François (1996, 2005). *SATO 4, Manuel de référence*, Centre ATO, UQAM, Montréal.
http://www.ling.uqam.ca/sato/satoman-fr.html

Duchastel Jules et al. (2005), ATONET, Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur.
http://www.atonet.net

ISO/TC 37/SC 4, *Language Resources Management, Morpho-syntactic Annotation Framework (MAF)*, ISO/CD 24611.
http://tc37sc4.org/

Heiden, Serge (2002). *Interface hypertextuelle à un espace de cooccurrences : implémentation dans Weblex*
http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2004/pdf/JADT_055.pdf

Lebart, L.(2005); *Data and Text Mining.* École nationale supérieure de télécommunications, Paris.
http://www.enst.fr/egsh/lebart/

Reinert, Max (2002). *Alceste, Manuel de référence*, Université de Saint-Quentin-en-Yvelines, CNRS.

Salem, André et al.(2003). *Manuel Lexico 3*, version 3.41.
http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/lexicoWWW/team.htm

The TEI Consortium (2005). *Text Encoding Initiative, The TEI Guidelines, P5, Guidelines for Electronic Text Encoding and Interchange,* edited by C.M. Sperberg-McQueen and Lou Burnard, Oxford, Providence, Charlottesville, Bergen.
http://www.tei-c.org/

Viprey, Jean-Marie (2005). *DiaTag -Astartex*, Université de Franche-Comté
http://laseldi.univ-fcomte.fr/document/viprey/page_JMV.htm

World Wide Web Consortium (W3C),
http://www.w3.org/