Les services Web pour un développement coopératif d'outils logiciels en analyse textuelle

François Daoust[1]

[1]UQAM – Montréal – Canada

Abstract

In this paper, we discuss the concept of Web service as a software architecture facilitating cooperative development and functional interoperability between independent modules of textual analysis. This reflection is in the wake of efforts within the network ATONET (http://www.atonet.net) to stimulate technological convergence in the field of textual analysis. In its most general sense, one can see the Web service as a software implementation of a computing resource, identified by a URI (Universal Resource Identifier), accessible via Internet protocols. This paradigm provides a standardized method of interoperability between software modules, regardless of platforms and programming languages used. Various application interfaces can thus call on these services in a transparent way without conditioning the development and the use of it, thus supporting a free and respectful development practices and expert testimonies of each one. This approach, based on a formalization of XML data formats, has also the advantage of stimulating our efforts of abstraction so that the data exchange formats are sufficiently general to make them sustainable and independent of specific constraints application software.

Résumé

Dans cette communication, nous voulons approfondir la notion de service Web à titre d'architecture logicielle facilitant le développement coopératif et l'interopérabilité fonctionnelle entre modules indépendants d'analyse textuelle. Cette réflexion se situe dans la foulée des efforts entrepris au sein du réseau ATONET (http://www.atonet.net) en vue de stimuler la convergence technologique dans le domaine de l'analyse de corpus textuels. Dans son sens le plus général, on peut voir le service Web comme l'implémentation logicielle d'une ressource de calcul, identifiée par un URI (Universal ressource Identifier), accessible en utilisant les protocoles Internet. Ce paradigme offre une méthode standardisée d'interopérabilité entre modules informatiques, indépendamment des plateformes et des langages informatiques utilisés. Diverses interfaces applicatives peuvent donc faire appel à ces services de façon transparente sans en conditionner le développement et l'utilisation, favorisant ainsi un développement libre et respectueux des habitudes et des expertises de chacun. Cette approche, basée sur une formalisation des formats de données en XML, a aussi l'avantage de stimuler nos efforts d'abstraction de telle sorte que les formats d'échange des données soient suffisamment généraux pour les rendre pérennes et indépendants des contraintes spécifiques des logiciels d'application.

Mots-clés : interopérabilité, interoperability, architecture informatique, software architecture, service Web, Web service, XML, TEI.

1. Introduction

La communauté des concepteurs de logiciels d'analyse textuelle se recrute encore largement dans le milieu de la recherche universitaire dont les moyens, on le sait, sont limités. Qui plus est, la nature pluridisciplinaire de l'analyse de discours requiert inévitablement une multitude d'outils et d'interfaces ouvertes d'analyse de corpus. L'idée d'un logiciel unique répondant à tous les besoins est donc ici, encore plus qu'ailleurs, difficilement imaginable. C'est donc plutôt du côté de la collaboration entre concepteurs et de l'interopérabilité entre outils spécialisés que se pose, d'après nous, la question du développement d'environnements logiciels aptes à servir la communauté des analystes de corpus. Dans cette communication, nous voulons discuter de l'intérêt du paradigme des *services Web* comme voie possible pour faciliter ce développement coopératif souhaité.

Dans un premier temps, nous voulons revenir sur la problématique de l'interopérabilité telle qu'elle se pose dans le contexte de plus en plus dominant de l'architecture Web. Dans un deuxième temps, nous voudrions faire état de nos expériences de développement utilisant cette architecture pour explorer, dans un troisième temps, un format TEI pour représenter des contextes issus d'opérations de concordance.

2. Contexte

Le problème de l'interopérabilité entre modules de traitement informatique se posait déjà avec suffisamment d'acuité en 2005 pour que l'ATALA en fasse l'objet d'une journée d'étude[1]. Suivant une proposition de formation de groupes de travail JADT, le réseau de collaboration ATONET formera, dès sa fondation en 2005, le *Groupe sur les formats d'échange de documents électroniques*. C'est dans le cadre du travail de ce groupe qu'ont été formulées des propositions d'utilisation d'un noyau restreint de balises TEI (*Text Encoding Initiative*) destiné à produire des documents TEI servant de format pivot (Daoust et Marcoux 2006) pour l'échange de corpus utilisant au départ les *formats propriétaires* de logiciels connus de textométrie et d'ATO : Alceste (Reinert, 2002), DTM (Lebart, 2005), Lexico (Salem *et al.*, 2003) et SATO (Daoust, 1996, 2011).

Dans ces propositions, dites de *Sacacomie*[2], le *format élargi* proposait l'utilisation de la balise *w* (*word*) pour rendre compte du découpage du flux textuel en occurrences (*token*). L'identification de chacune de ces occurrences au moyen de l'attribut standard *iml:id* permet de référer à des mots déterminés ou à des empans textuels à des fins d'annotation et d'analyse. Il peut s'agir d'annotations simples concernant une unité ou d'annotations structurelles mettant en relation plusieurs unités.

Cette proposition de format d'échange normalisé de corpus doit cependant être complétée par des propositions plus spécifiques permettant d'échanger des vues construites sur ces corpus

1 *Articuler les traitements sur corpus* organisée à Paris par l'Association pour le Traitement Automatique des Langues (ATALA)

2 Sacacomie est le nom du lac où s'est tenu le séminaire ATONET où furent présentées et discutées ces propositions de format d'échange de corpus. Deux programmes en *Perl* ont été développés pour agir comme passerelle de conversion de chacun des formats propriétaires au format pivot et du format pivot vers chacun des formats propriétaires.

au-delà des représentations internes adoptées par chacun des logiciels d'analyse textuelle. D'où l'idée, formulée au sein d'ATONET, de travailler à la modélisation de formats d'échange des *données intermédiaires* impliquées dans les chaines de traitement de l'ATO. Sur la base de ces formats, il deviendrait possible d'appeler directement des fonctions de calcul développées par les chercheurs sans devoir passer par l'interface native des logiciels qui les implantent. Cette séparation entre l'interface usager et la fonction de calcul, à laquelle on accède au moyen d'un protocole de communication et d'un format d'échange des données, est donc vue comme un moyen de faciliter la mise en commun et l'utilisation de nos méthodes et fonctions d'analyse. C'est dans ce contexte qu'un premier effort de normalisation XML-TEI de *données intermédiaires* a conduit à l'implantation d'analyseurs de cooccurrences sous la forme de services Web, effort que nous voulons prolonger ici en examinant un format XML-TEI pour des contextes produits par des mécanismes de type concordance.

3. Services Web : thème et variations

Dans son sens le plus général, on peut voir le service Web comme l'implémentation logicielle d'une ressource de calcul, identifiée par un *URI (Universal ressource Identifier)*, accessible en utilisant les protocoles Internet.

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML. They can be combined in a loosely coupled way in order to achieve complex operations. Programs providing simple services can interact with each other in order to deliver sophisticated added-value services. (http://www.w3.org/2002/ws/Activity)

Cette définition générale des services Web par le W3C met en évidence la notion d'interopérabilité d'applications logicielles indépendamment du langage utilisé pour les programmer, de l'architecture des calculateurs et des systèmes d'exploitation gérant ces calculateurs. Les requêtes, avec leurs données et leurs résultats, peuvent s'exprimer en utilisant diverses syntaxes concrètes, mais elles ont en commun de circuler à travers un réseau utilisant les standards du Web, en particulier l'URI[3] qui correspond à l'idée courante d'*adresse Internet*.

Il existe une norme permettant de décrire un service Web. Il s'agit de *Web Services Description Language*[4] (WSDL). Comme l'indique le document du W3C...

> ... a WSDL 2.0 document is inherently only a *partial* description of a service. Although it captures the basic mechanics of interacting with the service — the message types, transmission protocols, service location, etc. — in general, additional documentation will need to explain other application-level requirements for its use. For example, such documentation should explain the purpose and use of the service, the meanings of all messages, constraints on their use, and the sequence in which operations should be invoked.

3 Une extension aux URI, appelée IRI, a été proposée pour supporter l'internationalisation. IRI [IETF RFC 3987]. Voir http://www.w3.org/International/articles/idn-and-iri/

4 Pour plus d'information, voir Web Services Description Language (WSDL) Version 2.0 Part 0: Primer http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626

Un des modèles d'implantation de services Web est basé sur l'architecture REST (*Representational State Transfer*), d'après le terme inventé par Roy Fielding en 2000. Cette architecture fait largement appel au protocole HTTP couramment utilisé par les internautes dans leur navigation quotidienne.

Au Centre d'analyse de texte par ordinateur de l'UQAM, cette approche a commencé à être utilisée dès le milieu des années 1990 dans le cadre de l'interface Web utilisée pour l'implantation *client-serveur* du logiciel SATO (Daoust 1996, 2011). Cette technologie, écrivions-nous en 2004, *permet d'envisager un véritable travail coopératif jumelant un espace de travail personnel avec des ressources partagées : corpus, bases de données lexicales, documentation et guides méthodologiques* (Duchastel et coll. 2004).

Sur le plan informatique, ce choix d'une architecture reposant sur les protocoles du Web permet de séparer la couche interface de la partie proprement algorithmique du logiciel. Elle permet de combiner des traitements autonomes, rendus accessibles grâce à une interface standard indépendante du logiciel et des *API propriétaires*. Pour réaliser cette implantation, on a développé un service Web générique[5] qui agit à titre de passerelle entre les applications de calcul et le navigateur Web de l'utilisateur, ou tout autre programme accédant au service par le protocole HTTP. La passerelle permet de démarrer un programme ou de transmettre des commandes à un programme déjà en exécution. Le service passerelle retourne un fichier qui pourra incorporer les résultats produits par un programme démarré par la passerelle. Des instructions de traitement inscrites dans ce fichier servent à contrôler le dialogue avec les programmes qui tournent dans l'espace de travail de la session. Du point de vue de l'interface, une session de travail prend donc la forme d'un enchainement d'appels à la passerelle qui actualisent des pages Web servant de gabarit. Les fichiers HTML retournés par la passerelle contiennent des hyperliens et bordereaux HTML qui permettront de rappeler à nouveau la passerelle et de contrôler ainsi les prochaines étapes de la session de travail. Aussi, par l'utilisation des fenêtres multiples (cadres), l'utilisateur se verra offrir un ensemble d'hyperliens permettant de contrôler sa session de travail. Ce modèle correspond bien au fonctionnement de l'architecture REST telle que présentée par Roy Fielding.

> Representational State Transfer is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use. (Fielding, R. 2000)

L'utilisation de ce service Web agissant à titre de passerelle générique a l'avantage de libérer les logiciels d'application des tâches de validation de la session et de décodage des paramètres transmis par le protocole HTTP. Mais, on peut aussi concevoir des services Web qui répondent sans intermédiaire à une requête HTTP.

5 Pour plus d'information sur le fonctionnement de ce service, voir *Guide de programmation des interfaces HTML http://www.ling.uqam.ca/sato/satoman-fr_interface.html*

4. Le passage au format XML-TEI

Dans l'histoire de développement du logiciel SATO, il a été convenu de se concentrer sur un modèle de représentation et de traitement du corpus avec ses diverses couches d'annotation. Les traitements statistiques avancés ont donc été conçus dès le départ comme des modules externes opérant sur des données secondaires produites par SATO sous la gouverne de l'usager. C'est le cas de l'analyse des cooccurrences. SATO est utilisé pour bâtir des contextes et compter des objets dans ces contextes. Au départ, l'exportation de ces données prenait la forme d'un tableau délimité par des caractères de tabulation et des fins de ligne. Ces tableaux représentent les *matrices creuses* de façon peu économique puisque le nombre de tabulations est en fait supérieur aux données non nulles.

La première implantation du service Web de la cooccurrence remonte au début des années 2000 et ne faisait pas appel à la normalisation XML. Le tableau de données était donc transmis dans la forme traditionnelle dite CSV (*Comma-separated values*) utilisant cette variante qui remplace la virgule par la tabulation. Il s'agissait donc essentiellement à l'époque d'utiliser un service Web pour appeler un programme fonctionnant à la manière de la feuille de calcul autonome qu'on utilisait auparavant. Pour compléter le tableau de données transmis au moyen d'une référence au fichier texte qui le contenait, il suffisait de transmettre au service Web quelques paramètres généraux permettant de guider l'analyse. En retour, le service Web retournait, sous forme de tableau HTML, les cooccurrents significatifs accompagnés d'indices numériques.

Mais, les limites de ce format de données sont apparues lorsque nous avons voulu augmenter les fonctions du service Web pour implanter des algorithmes requérant des données supplémentaires. Plus fondamentalement, le passage à un format structuré s'est imposé comme une nécessité à partir du moment où nous voulions récupérer les résultats du calcul de cooccurrence sous forme d'annotations enrichissant le corpus et susceptibles d'être soumises à d'autres services Web.

L'utilisation de documents structurés permet d'organiser un flux de données en parties distinctes et en composantes clairement identifiées. Le langage de balisage XML constitue aujourd'hui la norme privilégiée pour constituer des documents structurés. Mais, encore faut-il convenir, au sein d'une communauté spécifique, de formats XML précis qui correspondent à un modèle de données adapté à une catégorie de traitements. L'objectif d'interopérabilité des traitements nous invite donc à un effort de généralisation et de normalisation pour l'établissement de formats de données bien définis, documentés et suffisamment génériques pour ne pas dépendre des particularismes du contexte applicatif. Dans la mouvance d'ATONET, nous avons convenu de s'inspirer des recommandations de la TEI pour établir de tels formats pour une communauté d'analyse de texte inscrite dans une tradition de sciences humaines.

Ce choix de privilégier l'utilisation d'un ensemble basique de balises XML-TEI se justifie aussi par le projet de constituer un dépôt de données pour les corpus et les documents qu'ils rassemblent de même que pour les documents d'analyse qui annotent les textes et les corpus (Daoust et coll., 2008). Dans ce contexte, on est susceptible de vouloir déposer des documents de cooccurrence au titre de documents d'annotation présentant une vue analytique sur le corpus dont il est issu. Au-delà du service Web de cooccurrence (Martinez, Daoust et Duchastel, 2010) que nous voulions perfectionner, c'est le statut documentaire du fichier de cooccurrence que nous visions à établir. Le fait de privilégier un même format TEI pour les textes, les documents d'annotation et d'analyse permet de créer des corpus de recherche TEI rassemblant, pour des fins

donnés, un ensemble de documents. Certes, des transformations sont possibles pour importer ou exporter des documents dans d'autres formats selon diverses exigences applicatives. Mais, la cohérence de l'ensemble du système sera facilitée par le partage à l'interne d'un format XML-TEI basique.

Pour l'implantation du dépôt de données, nous avons choisi d'utiliser le logiciel à code ouvert Fedora qui repose sur le concept d'objet numérique (Kahn, Wilensky, 1995, 1996) et sur une architecture faisant systématiquement appel aux services Web fonctionnant comme des méthodes permettant d'agir sur des types d'objets définis dans des modèles de données aussi considérés comme objets numériques. Dans ce contexte, la formalisation des formats d'échange des *objets de l'analyse textuelle* et la définition fonctionnelle de classes de traitement en termes de services Web agissant sur ces formats comportent aussi une dimension documentaire. Ainsi, on facilitera la diffusion des résultats de l'analyse textuelle informatisée au-delà des articles qui en font la synthèse.

Pour la cooccurrence, nous avons convenu de définir un modèle de données basé sur trois ensembles correspondant à des divisions au sens de la balise *div* de la TEI.

1. Un ensemble d'informations générales nécessaires à la compréhension des autres ensembles de données (*div type="Statistiques"*). On y trouvera notamment la description des requêtes dont les résultats se retrouveront dans les blocs suivants. Ce bloc de données fait partie du corps même du document de cooccurrence en ajout à l'entête TEI qui documente l'ensemble du document.
2. Un ensemble d'*objets*, avec leur description, dont on veut mesurer la cooccurrence statistique (*div type="objet"*).
3. Un ensemble de contextes qui définit les lieux du calcul de la cooccurrence (*div type="Contextes"*). Ce qui nous intéresse pour la cooccurrence, c'est de savoir quels sont les objets qui, d'après un certain modèle probabiliste, occurrent ensemble dans les contextes avec une fréquence difficilement explicable par le hasard.

Les objets sont définis par un ensemble de traits utilisant le formalisme des structures de traits[6] (*fs*). Voici un exemple de description d'objet extrait d'un document TEI soumis au service de calcul de la cooccurrence. Cet exemple contient aussi un résultat ajouté par le service de la cooccurrence.

6 À quelques variantes près, le formalisme de formalisme introduit par la TEI est repris sous forme de recommandations ISO.

```
<div type=»objet»>
<fs xml:id=»obj98» type=»Cooc» n=»98»>
   <f name=»Effectif»><numeric value=»92»/></f>
   <f name=»Contexte_Nbr»><numeric value=»90»/></f>
   <f name=»Description»><string>canadienne</string></f>
   <f name=»Id»><string>px3034</string></f>
   <f name=»Cooc» ana=»#req1» n=»contexte_bino»>
      <numeric n=»attendu» value=»1»/> <numeric n=»observé» value=»10»/> <numeric n=»prob»
value=»0.00000001»/> </f>
</fs> <!-- ... --> </div>
```

Tableau 1: Exemple de définition d'objet

La description des objets à dénombrer est contenue dans un élément TEI *div*. Chaque objet est décrit par une structure de traits utilisant un élément *fs* de type *Cooc.* À l'intérieur de l'élément *fs*, on trouve un ensemble de traits (éléments *f* pour *feature*). Les traits fournissent la valeur d'un certain nombre de propriétés de l'objet, nécessaires au calcul ou rajoutées par le service Web comme résultat du calcul. Chaque structure *fs* reçoit un identifiant unique (attribut *xml:id*) et un numéro s'il y a lieu.

Dans cet exemple, le trait *Effectif* indique le nombre total d'occurrences de l'objet dans le corpus (ou le sous-corpus). Le trait *Contexte_Nbr donne le nombre de* contextes où il apparait. Le trait *Id* renvoie à un identifiant de l'objet dans le corpus sur lequel porte le document de cooccurrence. Finalement, le trait *Description* contient un descriptif de l'objet, *canadienne* dans notre exemple. Ce trait optionnel n'est pas utilisé par le calcul et pourrait être omis pour plus de confidentialité dans le transport Internet du fichier. L'application appelant le service pourra utiliser la valeur de l'attribut *Id* pour référer à une description affichable de l'objet dans le corpus.

L'objet *canadienne* reçoit un trait supplémentaire qui a été ajouté par le service de cooccurrence en réponse à une requête. Dans l'exemple, le trait nommé *Cooc* est issu de l'analyse désignée par l'attribut *ana* dont la valeur pointe sur l'entrée *req1* dans la division *Statistiques*. Le contenu de la balise décrit un résultat significatif du point de vue de la loi binomiale appliquée au décompte des contextes.

Le deuxième ensemble de données définit les contextes en référence au corpus dans sa forme normalisée XML-TEI dans laquelle le découpage en mots correspond à une suite de balises TEI *w* comme l'illustre le tableau 2. Ce tableau montre une phrase issue d'un corpus sur le *corpus constitutionnel canadien 1941-1987* (Bourque et Duchastel, 1996).

```
<pb n=»dcc-1941/12»/>
<p><lb n=»40»/><w xml:id=»w6326»>On</w> <w xml:id="w6327">n'</w><w xml:id="w6328">y</w>
<w xml:id="w6329">relève</w> <w xml:id="w6330">aucune</w> <w xml:id="w6331">tentative</w>
<lb n="41"/><w xml:id="w6333">de</w> <w xml:id="w6334">bouleverser</w> <w
xml:id="w6335">la</w> <w xml:id="w6336">forme</w> <w xml:id="w6337">vraiment</w> <w
xml:id="w6338">canadienne</w>
<lb n="42"/><w xml:id="w6340">de</w> <w xml:id="w6341">notre</w> <w
xml:id="w6342">fédération</w><w xml:id="w6343">.</w> </p>
```

Tableau 2: Exemple de paragraphe avec balisage des mots dans le corpus dcc.xml

L'élément TEI span permet de référer à ce contexte comme l'illustre le tableau 3.

```
<div type=»Contexte» xml:base=»dcc.xml»>
  <span type=»Dénombrement-cb» from=»#w6326» to=»#w6343» xml:id=»w6326-w6343» n=»16»>
    <cb n=»98»/>1<cb n=»219»/>1<cb n=»472»/>1<cb n=»555»/>1<cb n=»578»/>1 </span>
</div>
```

Tableau 3: Désignation d'un contexte par l'élément span de type Dénombrement-cb

Dans ce format, le contexte est défini comme un empan textuel (*span*) allant des mots *w6326* à *w6343* référant aux attributs *xml:id* des balises *w* qui découpent le corpus en mots dans le fichier *dcc.xml* dont est extrait le Tableau 2. L'attribut *n* donne la longueur de l'empan textuel dans le corpus de référence. Cette longueur se calcule en nombre d'occurrences (mots). L'attribut *type=''Dénombrement-cb''* permet de guider l'interprétation du contenu de l'élément *span*.

Dans son format 2010, le service Web accepte deux modes de représentation des décomptes d'objets. Le premier mode est une transcription littérale de la matrice de données utilisée dans le format pré-XML. Le contenu du *span* est composé d'une suite de paires formées d'une balise vide *<cb/>* suivie d'un nombre correspondant au dénombrement d'un objet présent dans le contexte. La balise *cb* marque un début de colonne dans une ligne de texte. On utilise l'attribut *n* pour indiquer le numéro de la colonne. Cette idée de colonne est donc une traduction directe de la représentation des données sous forme de tableau dont les *span* formeraient les lignes. Dans ce format, chaque colonne contient le nombre d'occurrences de l'objet compté dans l'ordre séquentiel des objets définis. Il s'agit d'une *matrice creuse* composée d'un très grand nombre de zéro. Dans le format XML utilisé ici, on assume qu'une colonne qui n'est pas décrite contient zéro comme valeur implicite. On peut donc omettre de la représentation XML la grande majorité des colonnes.

L'utilisation de l'attribut *type* de l'élément *span* permet de guider l'interprétation du contenu de la balise et donc d'introduire des variantes. Ainsi, pour pallier le caractère trop implicite de l'interprétation du *format colonne,* on a introduit un type de contexte (*Dénombrement-measure*) qui renvoie à l'utilisation de la balise TEI *measure* plutôt qu'aux délimiteurs de colonnes comme l'illustre le tableau 5.

```
<div type=»Contexte»> <span type=»Dénombrement-measure" from="dcc.xml#w6326"
to="dcc.xml#w6343" xml:id="w6326-w6343" n="16">
  <measure quantity="1" ana="#obj98"  type="Contexte_OccNbr"/>
  <measure quantity="1" ana="#obj219"  type="Contexte_OccNbr"/>
  <measure quantity="1" ana="#obj472"  type="Contexte_OccNbr"/>
  <measure quantity="1" ana="#obj 555" type="Contexte_OccNbr"/>
  <measure quantity="1" ana="#obj578 " type="Contexte_OccNbr"/>
</div>
```

Tableau 4: Désignation d'un contexte par l'élément span de type Dénombrement-measure

On comprend que *quantity* donne le nombre d'occurrences dans le contexte de l'objet *obj98* par exemple. La mention du type facilite la lecture humaine de la donnée, mais pourrait être omise si aucun autre type de mesure est admissible dans le contexte.

5. Soumettre la concordance au calcul de la cooccurrence

Dans cette section, nous voulons illustrer davantage comment la discussion des formats d'échange pour les résultats produits par les logiciels d'analyse textuelle est un exercice de formalisation qui, s'il est requis pour la définition de services Web autonome, permet aussi de réfléchir sur ces objets au-delà de leur contexte applicatif immédiat. On a donc voulu poursuivre la réflexion sur la définition des formats admissibles au service de la cooccurrence en examinant un autre objet de l'analyse textuelle informatisée, c'est-à-dire la concordance.

Nous entendons par concordance toute liste de contextes construite à partir d'un patron sélectionnant des segments de texte en fonction d'une ou de plusieurs contraintes booléennes ou positionnelles sur le contenu des segments. Sans extrapoler outre mesure sur les services Web que l'on pourrait envisager pour l'analyse des concordances, on peut déjà convenir que le format *dénombrement d'objets* conçu spécifiquement pour la cooccurrence est insuffisant pour l'application de services Web qui s'intéressent à la séquentialité des objets, comme c'est le cas des concordances. Plusieurs des opérations classiques d'analyse des concordances opèrent en effet sur la séquence des objets : ordonnancement des contextes mettant en évidence des récurrences de chaines de mots, de catégories ou de syntagmes, décomptes de ces récurrences sous diverses formes, etc.

Si les objets impliqués dans les concordances n'étaient qu'une simple reprise de suites d'occurrences du corpus analysé, un retour au fichier corpus pourrait suffire pour reconstruire les segments de textes recouverts par les contextes. Mais, l'objectif ici est de généraliser le formalisme de telle sorte qu'il puisse s'appliquer à tout type de séquence contextuelle. Il peut s'agir de séquences de formes lexicales, ou de séquence de traits construits sur la surface du texte : lemmes, catégories grammaticales, syntagmes, séquences dans une structure discursive, etc. Il peut s'agir de séquences mixtes mariant des annotations diverses. Nous nous situons donc dans l'esprit des travaux de Bénédicte Pincemin qui, à partir d'une exploration des diverses interfaces d'exploitation de la concordance, pose la question : *quelle généralisation opératoire peut-on définir de la méthode des concordances* (Pincemin 2006) ? Cependant, ce que nous voulons discuter ici, c'est moins l'aspect opératoire et visuel que le modèle de données qui rendrait possibles ces diverses exploitations. À ce titre, nous retenons l'idée que les contextes constitués à titre de concordance sont composés de zones. Pour les concordances constituées

autour d'un mot pôle, on retient généralement un découpage en quatre zones : contexte gauche, pôle et contexte droit complété d'une zone documentaire donnant la référence du pôle dans le corpus et autres données contextuelles.

> La sélection du pivot se détaille comme une séquence de zones successives : autrement dit, le pivot n'est pas un bloc, mais il est structuré, et composé d'une suite d'éléments individuellement identifiables et potentiellement actifs pour la construction de la présentation des résultats. (Pincemin 2006)

En fait, cette construction de zones guidant l'analyse des concordances n'a pas à se limiter à la position pôle lorsque le patron de concordance spécifie des contraintes sur plusieurs positions dans la séquence. Dans la terminologie du logiciel SATO (Daoust 1996, 2011), on emploie le terme *patron de concordance* pour désigner ce jeu de contraintes booléennes et positionnelles. Pour illustrer, considérons une construction syntaxique à degré variable de figement comme *prendre plaisir*. Entre le verbe et le nom, on peut trouver certaines séquences de mots (*prendre grand plaisir*, *prendre un malin plaisir*) pouvant constituer une ou plusieurs zones à explorer ou à exclure, sous la gouverne de paramètres transmis au service Web. Il pourrait s'agir, par exemple, du calcul de segments répétés (Lebart et Salem 1994) ou de *quasi-segments* (Becue et Peiro, 1993). Un document de concordance devrait donc, pour répondre aux besoins déjà identifiés, comprendre les éléments suivants.

- Une description du patron de concordance et de chacun des filtres qui le compose ;
- Une suite de contextes référant aux occurrences du corpus qui en délimitent les frontières de gauche et de droite ;
- Pour chaque contexte, on devrait avoir l'option de fournir des éléments supplémentaires de référence pour une compréhension autonome des spécificités du contexte : pagination, références structurelles, etc. ;
- Pour chaque contexte, on a la séquence des éléments que l'on veut soumettre à l'analyse, formes lexicales, catégories ou structures avec des pointeurs identifiant les occurrences du corpus recouvertes par l'élément ;
- Pour chacun des éléments, on voudrait aussi avoir un pointeur vers l'item du patron de concordance sélectionnant l'élément ;
- Si nécessaire, il faudrait pouvoir disposer d'un marquage explicite de zones pertinentes pour l'analyse.

La production de ces contextes structurés implique probablement une complexification du patron de concordance qui, en plus du repérage des contextes, déposerait des ancrages de zones. À titre d'hypothèse de travail, voici (tableau 5) un format TEI qui pourrait être considéré pour représenter ces informations.

<ab type=»Documentation» >dcc-1941/12 ligne 40-42</ab>
<milestone type="zone" n="1"/><ref target="dcc.xml#w6326" ana="#filtre1 #Gram_Pronom">***On***</ref> <ref target="dcc.xml#w6327">**n'**</ref><ref target="dcc.xml#w6328">**y**</ref> <ref target="dcc.xml#w6329">**relève**</ref> <ref target="dcc.xml#w6330">**aucune**</ref> < ref target="dcc.xml#w6331">**tentative**</ref>
<ref target="dcc.xml #w6333">**de**</ref> <ref target="dcc.xml #w6334">**bouleverser**</ref> <ref target="dcc.xml #w6335">**la**</ref> <ref target="dcc.xml #w6336">**forme**</ref> <milestone type="zone" n="2"/> <ref target="dcc.xml #w6337" ana="#filtre2 #Gram_Adverbe">**vraiment**</ref> <milestone type="zone" n="3"/> <ref target="dcc.xml #w6338"ana="#filtre3 #Lemme_Canadien">**canadienne**</ref>
<milestone type="zone" n="4"/> <ref target="dcc.xml #w6340">**de**</ref> <ref target="dcc.xml #w6341">**notre**</ref> <ref target="dcc.xml #w6342">**fédération**</ref><ref target="dcc.xml #w6343">**.**</ref>

Tableau 5: Désignation d'un contexte par l'élément span de type Concordance

Dans l'exemple du tableau 5, on utilise les éléments *ref* de la TEI pour lister les références aux mots du contexte identifiés par un pointeur vers le corpus fourni par l'attribut *target*. L'élément *ref* contient, à titre documentaire, la graphie du mot référé ou toute information pertinente sélectionnée, par exemple le lemme ou une valeur de propriété du mot. Il serait aussi possible d'utiliser l'élément vide *ptr* (*pointer*) pour référer à l'occurrence dans le corpus sans ajout de contenu. Les valeurs de l'attribut *ana* (analyse) sont des pointeurs qui nous dirigent vers des éléments analytiques. Il y a un pointeur vers le filtre sélectionnant le mot dans le patron de concordance. Par exemple, *filtre1*, correspondant à une occurrence de *On* dans notre exemple, pourrait décrire un filtre sélectionnant un pronom. Il peut y avoir d'autres pointeurs d'analyse. G*ram_Pronom,* par exemple, pourrait référer à une structure de trait pour la catégorie grammaticale *pronom* avec sa fréquence dans le corpus. *Lemme_Canadien* ferait référence à l'analyse de l'occurrence de *canadienne* à titre de lemme *Canadien.* Ces référence*s,* si elles sont consignées dans la division *Objet* du document de concordance, pourraient être admissibles à notre service de cooccurrence.

S'il est nécessaire de spécifier des zones, le patron de concordance pourrait ajouter des balises vides <*milestone*/> indiquant des frontières de zone. On utilise des balises vides plutôt que des balises structurelles pour éviter des incohérences d'emboitement dans le cas où des éléments de structure préexistants seraient aussi transmis, une analyse en composants syntaxiques par exemple. La fonction des zones, ici désignées par un numéro, devrait être spécifiée lors de l'appel d'un service sensible à ce paramètre.

Dans cette proposition, nous avons choisi d'utiliser un bloc anonyme (*ab)* pour transmettre un contenu informatif sur le contexte, par exemple la référence à l'édition imprimée du texte. La structure générale de ce document de concordance pourrait donc servir à la fois à un service de cooccurrence et à un service d'analyse de concordances.

6. Conclusion

Dans cette communication, nous avons voulu montrer que les services Web peuvent être utilisés comme méthode de conception facilitant le développement modulaire et coopératif d'outils logiciels en analyse textuelle. Cette approche vise surtout l'implantation d'algorithmes d'analyse

en aval des logiciels de gestion et d'annotation des corpus. Ces modules informatiques peuvent donc être développés indépendamment des interfaces usagers tout en pouvant être appelés depuis ces interfaces grâce aux protocoles normalisés du Web.

De fait, les *services Web* font déjà largement partie de l'utilisation quotidienne des ordinateurs, des téléphones intelligents et autres dispositifs intégrant des ordinateurs reliés à Internet. Cette autonomie de développement que permettent les services Web, combinée à l'utilisation de protocoles largement répandus d'échange de ressources numériques, créent des conditions de collaboration très favorables au sein de collectivités de développeurs et d'utilisateurs.

L'architecture REST, qui fait reposer l'interopérabilité sur la définition du format des ressources, a l'avantage méthodologique d'établir la discussion sur le contenu plutôt que sur les protocoles de messages entre programmes informatiques. Sans nier que la syntaxe concrète des documents échangés pose certaines difficultés techniques, dans l'ensemble, l'utilisation d'XML et de la TEI correspond de plus en plus à une culture technologique accessible. Le choix que nous faisons d'utiliser un ensemble restreint de balises ayant un caractère générique a pour objectif de faciliter la formalisation algorithmique et la simplicité de l'implantation.

La plupart des environnements de programmation donnent accès à des librairies de procédures permettant le décodage de structures XML et de paramètres transmis selon les normes du W3C. L'apprentissage de ce mode de développement ne pose donc pas trop de difficultés. Il faut aussi rappeler que l'utilisation de ces protocoles Web n'implique pas que les applications doivent transiter par le réseau Internet. De fait, les ressources soumises au traitement peuvent résider même sur un ordinateur personnel et s'échanger sans passer par le réseau et sans référence obligée à l'*infonuagique*[7]. Donc, les services Web peuvent être déployés à diverses échelles et selon une variété d'architectures matérielles.

Cette approche répond aussi à une dimension documentaire trop souvent négligée avec pour conséquence d'hypothéquer le suivi de la recherche et le partage des matériaux de la recherche au sein de la communauté scientifique. L'adoption du format TEI contribue à donner un statut documentaire à ces fichiers construits en cours d'analyse et qui possèdent une valeur au-delà du logiciel d'application qui les a produit. Comprenons que le format des fichiers d'échange est bel et bien un format d'échange et n'implique pas son adoption en tant que format interne des données pour le traitement. Il s'agit d'un format documentaire qui permet non seulement l'interopérabilité, mais aussi la discussion scientifique sur les procédures et leur validation.

Références

ATONET. *Réseau pour l'échange de ressources et de méthodologies en analyse de texte assistée par ordinateur*. http://www.atonet.net

Becue M., Peiro R. (1993) - Les quasi-segments pour une classification automatique des réponses ouvertes, in Actes des 2ndes Journées Internationales d'analyse des données textuelles, (Montpellier), ENST, Paris, p 310-325.

7 Ce néologisme proposé par l'Office de la langue française du Québec est un des équivalents au *cloud computing* anglophone. Cette référence à l'*informatique virtuelle,* pour reprendre une expression utilisée par l'entreprise Intel, renvoie aussi à une réalité de plus en plus omniprésente recouvrant tant le dépôt de fichiers sur des serveurs distants que les services Web, comme le courrier électronique et les suites bureautiques Web.

Bourque G.et Duchastel J., avec la collaboration de Armony, Victor (1996). *L'identité fragmentée: nation et citoyenneté dans les débats constitutionnels canadiens.* Montréal, Fidès, 375 p. Description du corpus : http://www.chaire-mcd.uqam.ca/ato-mcd/

Daoust, F. ; Marcoux, Y. ; Viprey, J.-M. (2010). L'annotation structurelle. *JADT 2010*. http://www.ledonline.it/ledonline/JADT-2010/allegati/JADT-2010-1145-1156_057-Daoust.pdf

Daoust et coll. (2008). Daoust, F.; Duchastel, J.; Marcoux, Y.; Rizkallah, E. Pour un modèle de dépôt de données adapté à la constitution de corpus de recherche, in *Actes des JADT-2008*, vol. 1, pp- 355-367, Presses universitaires de Lyon, 2008. ISBN 978-2-7297-0810-8. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2008/pdf/daoust-duchastel-marcoux-rizkallah.pdf

Daoust F. et Marcoux Y. (2006). Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés, in *Les Cahiers de la MSH Ledoux n° 3, Actes des JADT-2006*, vol. 1, pp 327-340, Presses universitaires de Franche-Comté, 2006. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2006/PDF/029.pdf

Daoust, F. (1996, 2011). *SATO 4, Manuel de référence*, Centre d'analyse de texte par ordinateur, UQAM, 2007; modifié en 2011. Montréal. http://www.ling.uqam.ca/sato/satoman-fr.html

Duchastel, J. ; Daoust, F. Della Faille, D. SATO-XML: une plateforme Internet ouverte pour l'analyse de texte assistée par ordinateur. In *Le poids des mots*, Actes des JADT-2004, vol. 1, pp- 353-363, Presses universitaires de Louvain, 2004.

Fedora. http://fedora-commons.org/

Fielding, T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Thèse de doctorat, University of California. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Kahn, R.; Wilensky, R. (1995). *A Framework for Distributed Digital Object Services*, Corporation for National Research Initiatives, University of California at Berkeley. http://www.cnri.reston.va.us/k-w.html

Kahn, Robert & Wilensky, Robert. "A framework for distributed digital object services"; *International Journal on Digital Libraries* (2006) 6(2). [doi:10.1007/s00799-005-0128-x]. http://www.doi.org/topics/2006_05_02_Kahn_Framework.pdf .

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

Lebart, L. et Salem, A. (1994). Statistique textuelle. Dunod, Paris.

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

Martinez, W. ; Daoust, F. ; Duchastel, J. (2010) Un service Web pour l'analyse de la cooccurrence. *JADT 2010*. http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt2010/allegati/JADT-2010-1079-1090_081-Martinez.pdf

Pincemin B. (2006) - « Concordances et concordanciers -De l'art du bon KWAC », Corpus en Lettres et Sciences sociales : des documents numériques à l'interprétation, Actes du XVIIe Colloque d'Albi Langages et Signification, Albi, 10-14 juillet 2006, Carine Duteil-Mougel & Baptiste Foulquié (éds), ISBN 2-907955-12-18, pp. 33-42, et Texto! [en ligne http://www.revue-texto.net/1996-2007/Parutions/Livres-E/Albi-2006/pincemin.pdf ISSN 1773-0120], juin 2006, vol. XI, n°2.

Pincemin, B. *et al.* (2006b) Concordanciers : Thème et variations. *Actes des 8es Journées internationales d'Analyse statistique des Données Textuelles* (JADT 2006), Jean-Marie VIPREY *et al.* (éds), Besançon : Presses Universitaires de Franche-Comté, ISBN 2.84867130.0, vol. II, pp. 773-784.

TEI Consortium (2007). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium, eds. http://www.tei-c.org/Guidelines/P5/