TXM : Une plateforme logicielle open-source pour la textométrie - conception et développement

Serge Heiden, Jean-Philippe Magué, Bénédicte Pincemin

TXM : Une plateforme logicielle open-source pour la textométrie – conception et développement

Serge Heiden, Jean-Philippe Magué, Bénédicte Pincemin

ICAR, Université de Lyon – ENS-LSH – 15 parvis René Descartes – B.P.7000 – F69342 Lyon cedex 07 – France

Abstract

The research project *Federation and Research Developments in Textometry around the creation of an Open-Source Platform* distributes its XML-TEI encoded corpus textometric analysis platform online.
The design of this platform is based on a synthesis of features of existing textometric software. It relies on identifying the open-source software technology available and effectively processing digital resources encoded in XML and Unicode, and on a state of the art of open-source full-text search engines on structured and annotated corpora.
The architecture is based on a Java toolkit component articulating a search engine (IMS CWB), a statistical computing environment (R) and a module for importing XML-TEI encoded corpora.
The platform is distributed as an open-source toolkit for developers and in the form of two applications for end users of textometry: a local application to install on a workstation (Windows or Linux) and an online web application.
Still early in its development, the platform implements at present only a few essential features, but its distribution in open-source already allows an open community development. This should facilitate its development and integration of new models and methods.

Résumé

Le projet de recherche *Fédération des recherches et développements en textométrie autour de la création d'une plateforme logicielle ouverte* diffuse sa plateforme d'analyse textométrique de corpus XML-TEI en ligne.
La conception de cette plateforme repose sur une synthèse des fonctionnalités des logiciels de textométrie existants. Elle s'appuie sur le recensement des technologies logicielles open-source disponibles et efficaces pour manipuler des ressources numériques XML et Unicode, et sur un état de l'art des moteurs de recherche en texte intégral sur corpus structurés et étiquetés.
L'architecture consiste en une boîte à outils Java articulant un composant moteur de recherche (IMS CWB), un environnement de calcul statistique (R) et un module d'importation de corpus XML-TEI.
La plateforme est diffusée sous la forme d'une boite à outils en open-source pour les développeurs informatique mais également sous la forme de deux applications pour les utilisateurs finaux de la textométrie : une application à installer sur un poste local (Windows ou Linux) et une application web accessible en ligne.
Encore au début de son développement, la plateforme n'implémente à l'heure actuelle que quelques fonctionnalités essentielles, mais sa diffusion en open-source autorise un développement communautaire ouvert. Cela doit faciliter son évolution et l'intégration de nouveaux modèles et méthodes.

Keywords: textometry, open-source, full text search engine, statistical analysis, xml-tei, natural language processing, eclipse rcp, grails framework

1. Introduction

Le développement de la lexicométrie a beaucoup bénéficié à ses débuts des efforts de mutualisation des programmes informatiques de statistiques portés par les associations de

statisticiens programmeurs comme l'ADDAD et le CESIA (Fénelon, 1981)[1]. Quand nous avons créé le mot *textométrie* dans les années 2000, pour rendre compte de l'évolution des unités sur lesquelles pouvaient porter les calculs, avec les développements du Traitement Automatique des Langues naturelles (TAL), et surtout de la diversité et de la qualité des représentations informatiques de toutes formes de structures textuelles sur divers médias (papier, web…), s'est également posée la question d'un effort de mutualisation des développements informatiques autour de la textométrie en tant que telle. Le projet financé par l'Agence Nationale de la Recherche française (ANR) appelé « Textométrie » <http://textometrie.ens-lsh.fr>, dont cet article rend compte des premiers résultats logiciels, a été lancé pour lever deux nouveaux verrous de la communauté d'analyse statistique des données textuelles :

- mettre en place une plateforme logicielle mutualisée par le biais du développement en mode open-source, en s'appuyant sur l'expérience des concepteurs de logiciels de lexicométrie ou de textométrie ;
- se donner les moyens de pouvoir traiter des ressources textuelles sous leur état technologique le plus avancé et standard possible en visant concrètement à pouvoir traiter des corpus représentés en XML-TEI[2] ayant éventuellement été enrichis avec des outils de TAL.

1.1. Objectifs

Les objectifs informatiques du projet ANR sont triples. D'une part construire une nouvelle plateforme commune composée d'une boîte à outils (toolbox) offrant les services de base pour produire des applications de textométrie, et deux applications prototypes : une locale, à déployer sur les postes des utilisateurs de Sciences humaines et Sociales (SHS) : Windows, Linux et Mac OS X[3], et une basée sur le web en mode client/serveur afin de permettre des analyses collaboratives par Internet. D'autre part faire en sorte que les applications soient accessibles et déployables le plus facilement possible par les utilisateurs SHS. Et enfin, s'assurer de performances suffisantes pour des ressources textuelles de grandes dimensions et complexité[4].

2. Spécifications et conception de la plateforme logicielle

Le cycle naturel d'un projet de développement informatique suit la séquence : spécification de fonctionnalités appliquées à des données définies, conception formelle, implémentation logicielle, puis recette à base de tests type pour vérifier que l'implémentation correspond aux spécifications. Dans le cadre de ce projet, les travaux se sont notamment appuyés sur (Pincemin, 2004) et (Heiden, 2006), puis, les phases de spécification et de conception ont été fusionnées pour aboutir à un modèle conceptuel de travail sur la base duquel une première

[1] On trouvera dans cet ouvrage une description critique très intéressante de la situation informatique dans les années 80 pour les statisticiens français développant ce qu'on appelait l' « Analyse des Données » pages 204 à 219.

[2] Text Encoding Initiative (TEI, 2008).

[3] A ce jour, le déploiement sur Mac OS X n'a pas encore commencé.

[4] Corpus de centaines de million d'occurrences étiquetées et disposant de plusieurs niveaux d'annotation : phrase, paragraphe...

implémentation a été réalisée[5]. Le projet est alors entré dans un mode de développement à cycle court : en livrant fréquemment de nouvelles versions des applications qui tiennent compte le plus possible des retours des utilisateurs des versions précédentes.

2.1. Le modèle conceptuel de travail

Le modèle est défini de la façon suivante : une chaîne est une succession d'unités, donnant une segmentation (ou un découpage) du corpus homogène et complète (au sens où elle couvre tous les éléments définis et pertinents pour le niveau de description considéré). L'enjeu est que la chaîne soit représentative du corpus pour les calculs textométriques.

Un corpus est décrit par une ou plusieurs chaînes ou plans textuels. Les chaînes traduisent des structures syntagmatiques de tous ordres : différents plans textuels (corps du texte, discours direct, titres de sections, commentaires, citations…) ou différents paliers linguistiques de description (chaîne de caractères, chaîne de mots, chaîne de phrases, chaîne de paragraphes, chaîne de tours de paroles, chaîne de textes...) mais également des segmentations ou découpages alternatifs pour un même palier : par exemple, une chaîne pour une tokenisation par l'outil A et une autre chaîne pour une segmentation lexicale par la procédure B.

Les unités d'une chaîne peuvent être en relation de contextualisation avec les unités d'une autre chaîne : on représente notamment ainsi que telles unités sont contenues dans telle autre ou que telle unité contient telles autres. Les unités de différents plans ou paliers peuvent entretenir des relations d'emboîtement ou de chevauchement.

Les unités sont qualifiées selon des propriétés. Les propriétés sont typées (type qualitatif énuméré, ensemble de traits, chaîne de caractères, entier naturel, etc.). Si la propriété est définie pour l'unité, alors l'unité a une valeur de la propriété qui lui est associée.

Une sélection est un sous-ensemble d'unités.

Ce modèle conceptuel est implémenté dans le modèle opérationnel par la fonctionnalité d'importation de corpus de la plateforme. A ce jour, le modèle opérationnel consiste en une arborescence unique d'unités non enchâssées. La multiplicité des plans, paliers et segmentations est réalisée par le biais de différentes procédures d'importation créant autant de configurations, et de corpus « internes », que nécessaire.

2.2. Le modèle opérationnel de travail

Le modèle conceptuel est mobilisé pour les calculs via des paramètres généraux :

- le corpus ou le sous-corpus - sélection sur le corpus. Le sous-corpus permet typiquement de sélectionner un sous-ensemble de textes, d'écarter des éléments péritextuels. Il procède généralement par filtrage ou élimination d'unités contextualisant les unités de la chaîne d'analyse (par exemple, choix de textes dans le cas d'une chaîne d'analyse lexicale). Toutes les opérations s'appliquant aux corpus s'appliquent également aux sous-corpus. Dans le reste de l'article nous les désignerons donc par « corpus ».
- la chaîne d'analyse ;

[5] Ce modèle conceptuel est à la fois traditionnel et novateur : en conservant une architecture classique distinguant unités et propriétés, il n'implémente pas par exemple le modèle de texte de (Pincemin 2008) ; en revanche il prépare la gestion d'une pluralité de segmentations, ce qui ne se pratiquait pas jusqu'à présent.

- la partition active sur le corpus (pour des calculs contrastifs) ;
- le contexte vertical : utilisé pour la localisation des occurrences. Il précise le contenu et la forme des indications de localisation affichables pour une unité.
- la propriété d'analyse : les valeurs de la propriété d'analyse choisie définissent les types des unités, autrement dit, deux unités présentant la même valeur sur la propriété d'analyse sont considérées pour le traitement en cours comme deux occurrences d'un même type.
- la propriété d'affichage : les résultats peuvent être exprimés dans les valeurs d'une autre propriété que celle qui a permis de faire le calcul. Par exemple, une concordance peut être construite autour du verbe *aller* (focus), triée sur le temps puis la personne (propriétés d'analyse), et affichée sur la graphie (propriété d'affichage).

Si le calcul est focalisé, un à deux focus, définis par des sélections, peuvent être fixés :

  - le focus d'unités permet de centrer les traitements sur certaines unités (ex. pivot d'une concordance, pôle d'un calcul de cooccurrences). Lorsque le calcul ne fait pas intervenir de focus d'unités, il est dit *panoramique* ;
  - le focus de parties est le choix d'une ou de plusieurs parties dans la partition active ; les résultats sont alors donnés uniquement pour chacune de ces parties, bien que le calcul ait porté sur tout le corpus considéré.

En ce qui concerne les fonctionnalités, leur spécification est basée sur une synthèse de l'état de l'art et la mise au point d'une typologie unifiée, orientée utilisateur (Pincemin & al., 2010).

2.3. Place du TAL dans la plateforme

Dans l'état actuel de la plateforme, les enrichissements issus d'outils de TAL comme les étiqueteurs morphosyntaxiques-lemmatiseur sont pris en charge soit au sein même de la représentation des données avant leur importation dans la plateforme, soit appliqués aux sources pendant le processus d'importation dans la plateforme sous forme de scripts.

3. Positionnement informatique : Ouverture, Standard, Diffusion

En termes de développement informatique, l'ouverture des sources logicielles est un gage de mutualisation et de capitalisation important pour la communauté de recherche en textométrie. Elle offre des possibilités de comparaison entre algorithmes et donc d'amélioration, d'augmentation de fonctionnalités par le partage, d'adaptabilité du logiciel à d'autres contextes – en particulier pour les types de ressources textuelles traités, et enfin de durabilité par la transmissibilité des procédés pour leur réutilisabilité.

L'ouverture offre la possibilité de réutiliser des composants logiciels open-source déjà disponibles qui répondent à certains besoins du projet. La plateforme s'est donc construite à partir d'une base logicielle conséquente qu'il n'aurait pas été possible de construire au préalable dans un environnement fermé étant donnés les délais et les ressources disponibles.

L'ouverture impose de se donner des règles de développement et de conformité à des standards explicites de programmation et d'architecture. C'est pourquoi nous avons choisi de situer le cœur logiciel de la plateforme au sein du langage de programmation Java qui fait l'objet de l'écosystème logiciel industriel parmi les plus ouverts qu'est le Java Community

Process (JCP) et qui produit un important environnement de composants logiciels standard ouverts. Dans ce contexte, nous avons choisi la plateforme Eclipse RCP[6] pour développer l'application locale multiplateforme (Windows, Linux et Mac OS X), s'appuyant elle-même sur le standard « Open Services Gateway initiative Alliance » (OSGi) pour sa propre architecture, ainsi que la plateforme Grails pour développer l'application web s'appuyant elle-même sur le standard d'architecture d'application web Spring[7].

La situation du développement informatique a beaucoup évolué depuis les années 80 avec Internet et la mise en place d'écosystèmes de développement répartis dans le monde entier avec un support de code source ouvert. La première phase informatique de notre projet a été de déployer l'ensemble du développement de la plateforme, appelée TXM, sur un site web de développement open-source international ouvert à tous. Dans le développement open-source, il y a deux façon de déployer les sources : soit on attend d'abord qu'une masse critique du développement du logiciel ait été atteinte avant d'ouvrir le code à la communauté, soit on ouvre le code dès la première version déployable du logiciel même si ce dernier ne réalise qu'une partie infime de ses fonctionnalités prévues. Dans le projet Textométrie, nous avons choisi la deuxième voie. La communauté peut donc désormais déjà tester, analyser et s'approprier la version 0.4.5[8] de la plateforme TXM à l'adresse web :

https://sourceforge.net/projects/textometrie.

Les développeurs d'applications disposent d'un site dédié à l'adresse https://sourceforge.net/apps/mediawiki/textometrie/index.php?title=Main_Page documentant tout ce qui est nécessaire à l'appropriation des sources de la boîte à outils et du déploiement des applications.

Une plateforme d'hébergement comme Sourceforge offre d'importants services d'animation communautaire pour le développement logiciel.

Or, il nous semble remarquable de noter que le standard de représentation des corpus textuels en XML qu'est la TEI héberge l'animation de ses spécifications sur la même plateforme à l'adresse web https://sourceforge.net/projects/tei.

Cet usage conjoint de services d'hébergement et de développements ouverts à la fois pour la représentation des corpus textuels - ouverts ou open-source - et pour les logiciels - ouverts ou open-source -, censés les traiter, nous semble former une dualité intéressante conditionnant leur compatibilité d'évolution mutuelle.

4. Architecture

Le cœur de la plateforme TXM est sa boîte à outils (*toolbox*) intégrant les composants d'importation, d'édition, de recherche en texte intégral et de calcul statistique (voir la figure 1). Les composants de recherche en texte intégral et de calcul statistique communiquent avec la boîte à outils au moyen de *sockets* ce qui permet d'envisager des déploiements répartis sur différentes machines pour des environnement de production lourds.

[6] Pour « Rich Client Platform ».

[7] La plateforme Grails a également l'intérêt d'être développée en Groovy qui est le langage de script que nous avons choisi pour le scriptage de la plateforme TXM.

[8] Les chiffres composant le numéro de version indiquent le degré de maturité de chaque niveau de granularité de la plateforme : quand le premier chiffre est '1', la version correspond à l'objectif initial du projet - ce qui n'est pas le cas pour l'instant.

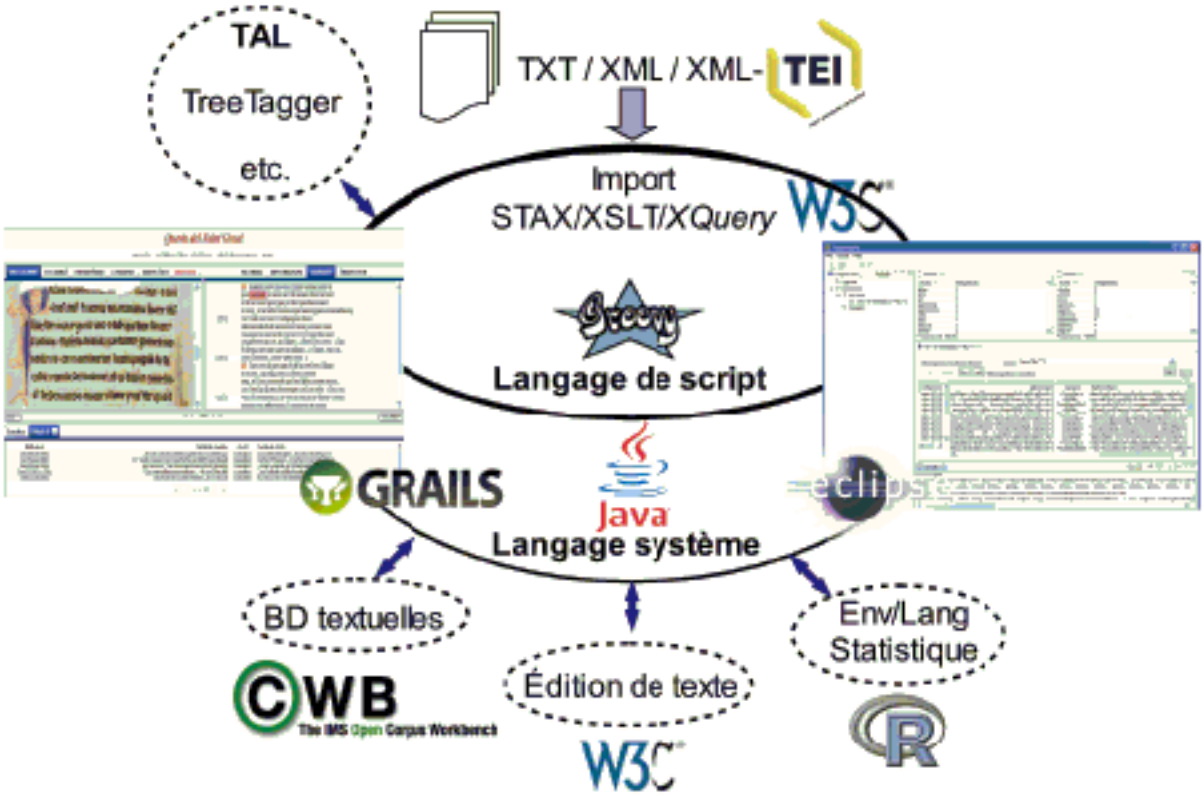

Figure 1 : architecture de la plateforme TXM – la boîte à outils est cerclée en trait plein, les composants cerclés de pointillés sont des processus indépendants. La copie d'écran de gauche correspond à l'interface de l'application web développée en Grails, la copie d'écran de droite correspond à l'interface de l'application locale développée en Eclipse RCP.

4.1. Description des composants

4.1.1. Composant « boîte à outils » de la plateforme

Le composant de base de la plateforme TXM est une librairie Java reliant les différents composants entre eux afin d'implémenter le modèle conceptuel et opérationnel. Elle expose ses fonctionnalités aux applications par le biais d'une API[9] documentée en ligne. Cette API est organisée en quatre parties principales : la première encapsule l'accès à la base de données textuelle et permet la manipulation des corpus (partitionnement, création de sous-corpus, requêtes sur les corpus…) ; la seconde encapsule l'accès à l'environnement de statistique et propose des structures de données sur lesquelles elle rend possible des calculs statistiques de bas niveau ; la troisième est de plus haut niveau et réunit les fonctionnalités des deux premières pour fournir des fonctionnalités propres à la textométrie (calcul des spécificités, concordance, analyse factorielle des correspondances...) ; enfin la quatrième est dédiée à l'importation et offre des fonctionnalités utilisées par les scripts d'importation.

4.1.2. Composant de recherche en texte intégral

La notion de *sélection* du modèle conceptuel a été implémentée à l'aide de technologies de moteurs de recherche. Le retour du moteur de résolution est une sélection utilisée pour construire les sous-corpus et définir les focus. Pour choisir la technologie open-source

[9]API : Application Public Interface.

adéquate[10], nous avons combiné la vérification de conformité maximale aux standards à l'analyse des fonctionnalités disponibles et à la réalisation de tests de performance sur deux niveaux possibles de représentation des données textuelles : les moteurs XML natifs et les moteurs de recherche textuels spécialisés. Les fonctionnalités privilégiées ont été celles relatives à l'accès et à la sélection efficace d'unités de granularité lexicale et à la construction et la gestion de sous-corpus. Dans ce contexte fonctionnel, le test de performance déterminant pour le choix de la technologie a été celui du repérage le plus rapide possible d'occurrences lexicales. Deux requêtes ont été testées, l'une cherchant un mot ayant une propriété particulière égale à une valeur donnée (par exemple toutes les occurrences du lemme « manger »), l'autre cherchant un mot ayant une propriété particulière reconnue par une expression régulière (par exemple toutes les occurrences dont la forme graphique est reconnue par l'expression régulière « .*prendr.* »). Les moteurs XML natifs testés ont été Sedna, DB XML et eXist-db. Les moteurs de recherche textuels testés ont été : IMS CWB, Manatee et Xaira[11].

À l'issue de ce banc d'essai, c'est la plateforme IMS CWB (Christ, 1994), qui a été choisie comme support de représentation interne des corpus de la plateforme TXM. Sur un corpus test de trois millions de mots, IMS CWB parvient à répondre à la première requête en moins de 1ms et à la deuxième en moins de 10ms[12]. Ce choix a hélas été fait au détriment (temporairement) de la conformité complète au standard d'encodage des caractères Unicode[13]. Pour l'expression d'un focus, IMS CWB offre un moteur de résolution de requêtes sous forme d'équations algébriques[14] parmi les plus reconnus et performants des bases de données textuelles. Ce potentiel, et la complexité associée, est offert selon différents degrés à l'utilisateur selon l'interface d'utilisation choisie. Dans tous les cas, l'interface minimale de recherche d'une forme graphique de mot reste l'opération la plus performante.

4.1.3. Composant de calcul statistique

Pour le choix du composant d'analyse statistique, la question principale a été celle de savoir si on privilégiait le choix d'un composant le plus proche possible du langage cœur de la

[10] La plateforme TXM étant diffusée en open-source, nous avons pu profiter de composants open-source déjà disponibles. Parmi tous les critères de sélection de tels composants, la qualité de la dynamique de l'écosystème des développeurs les produisant est un des plus difficiles à évaluer. Un composant peut par exemple être prometteur selon certains critères, par exemple fonctionnels ou de performance, mais voir son développement ralentir. Or un logiciel non maintenu posera toujours problème face à l'évolution des environnements et des standards informatiques. Il est donc important dans la conception de l'architecture de trouver un équilibre entre le potentiel d'un composant et son remplacement éventuel sans trop d'investissement pour le projet.

[11] Nous avons appris après notre campagne de tests que la configuration de Xaira que nous avons utilisée ne permettait pas de tirer pleinement parti de ses capacités.

[12] Sur un PC Pentium 4 à 3GHz avec 2,4Go de RAM sous Linux 2.6.

[13] La plateforme TXM gère totalement les corpus dont le système d'écriture est encodable en 8 bits (ISO/IEC 8859, 1998) et partiellement les corpus encodés en UTF-8 (Unicode, 2006). Le projet open-source IMS CWB, auquel nous participons, est en train de migrer vers la compatibilité Unicode complète à moyen terme, au bénéfice du projet Textométrie.

[14] Par exemple, la requête « `[lemma="love" & pos="V..i.*"]` » exprime la sélection de tous les verbes « to love » à l'imparfait dans un corpus lemmatisé et étiqueté avec le jeu morphosyntaxique Multext (Nancy, 1994).

plateforme (Java) plutôt que la richesse et la dynamique de la communauté de développement du composant. Il semble que le projet de langage fédératif (et donc open-source) pour les statistiques de (Fénelon, 1981) se soit réalisé dans l'environnement R (R, 2005), et c'est ce que nous avons finalement choisi bien qu'il ne soit pas Java. Avec le choix de ce composant, nous avons de bonnes garanties d'intégrer un composant qui bénéficiera des meilleures avancées en programmation statistique d'une communauté mondiale et diversifiée.

L'environnement R est par nature extrêmement modulaire : les fonctionnalités offertes par défaut sont limitées et de bas niveau, et ce sont des packages additionnels, développés essentiellement par la communauté des utilisateurs qui permettent d'enrichir l'environnement. Le package Rserve est ainsi utilisé pour permettre à R d'être accessible par *sockets*. De plus, nous avons développé un package, TextometrieR, qui offre toute une panoplie de méthodes statistiques propres à la textométrie (calcul des spécificités, des cooccurrences…). Ce package dépend lui-même de fonctionnalités offertes par d'autres packages : RSvgDevice pour la production de graphiques au format standard SVG (W3C, 2004) et CA pour réaliser l'analyse factorielle des correspondances. Outre la pertinence de leurs fonctionnalités, le choix de ces deux packages par rapport à d'autres offrant des services semblables, se justifie par le fait qu'ils ne dépendent pas à leur tour d'autres packages, limitant ainsi le nombre de dépendances de la plateforme.

4.1.4. Composant d'édition

Le composant d'édition intervient dans le cadre de la fonctionnalité de *retour au texte*. Il sert à restituer l'apparence d'un empan textuel donné (par exemple celle des occurrences correspondant au focus et de leur contexte immédiat). Actuellement, ce composant prend différentes formes selon les applications (locale ou en ligne) mais repose fondamentalement sur des composants de technologie HTML.

4.1.5. Composant d'importation

Dans la conception de ce composant, nous avons défini quatre périmètres principaux de qualité de représentation des données :

1. TXT : ce qui est moins ou différemment structuré que XML : texte brut, cvs (Excel), cnr (Cordial)... ;
2. XML : ce qui contient toutes les informations nécessaires codées en XML et les caractères en Unicode ;
3. XML-TEI pivot - qui contient toutes les informations nécessaires codées en TEI ;
4. IDX - la représentation interne des informations pour la plateforme TXM.

Toutes les opérations transformant des informations se trouvant dans des documents situés entre - et au sein - des trois premiers périmètres sont des opérations de *normalisation*. L'opération de création de la représentation interne à partir du XML-TEI pivot est l'opération de *compilation*. L'ensemble de ces opérations correspond à l'opération d'*importation* des sources dans la plateforme. Comme la figure 2 l'illustre, une importation peut prendre plusieurs voies et types de traitements possibles. Par ailleurs, nous nous sommes donné la possibilité de pouvoir appliquer des traitements de TAL pendant l'importation d'un corpus dans la plateforme. A ce stade initial des développements, étant donnée la multiplicité des cas de figure dans la procédure d'importation, et en prévision des évolutions futures du modèle conceptuel, nous avons préféré dans un premier temps porter l'effort sur la mise en place d'un

environnement de programmation de *procédures* d'importation simple plutôt que sur la définition d'un *format* d'entrée représentatif de ce que la plateforme TXM sera capable de traiter. Cet environnement est construit à partir du langage de script Groovy qui non seulement permet d'adapter facilement un script d'importation déjà existant pour prendre en charge un format de représentation légèrement différent, mais aussi de bénéficier des toutes les librairies Java disponibles[15]. Avec cet environnement, les procédures d'importation pourront non seulement s'adapter facilement aux évolutions de la TEI mais également prendre en charge un grand nombre de formats textuels d'entrée. Par ailleurs, les services Java d'intégration d'outils externes font bénéficier à cet environnement d'une extensibilité souple en TAL demandée par le projet.

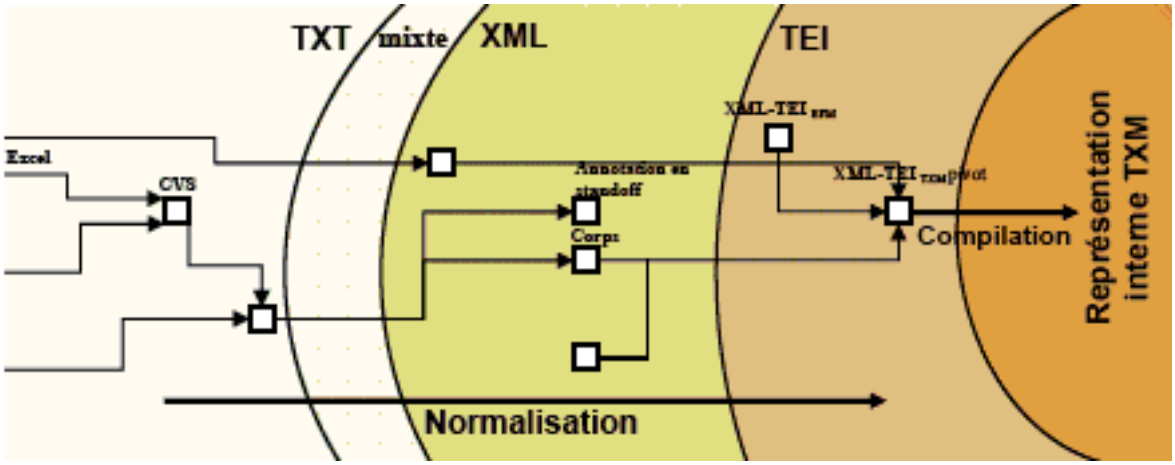

Figure 2 : opérations d'importation des données sources dans la plateforme TXM

5. Interfaces utilisateurs

Les figures 3 et 4 donnent un aperçu de l'interface Windows et Linux de la version 0.4.0 du prototype d'application locale. La figure 5 illustre l'interface de l'application web sur une édition critique de manuscrit médiéval (Marchello-Nizia et al, à paraître).

[15]Ce langage est en effet d'abord construit comme une surcouche syntaxique simplificatrice strictement compilée en Java avant son exécution. Le riche environnement Java est donc entièrement disponible dans les scripts.

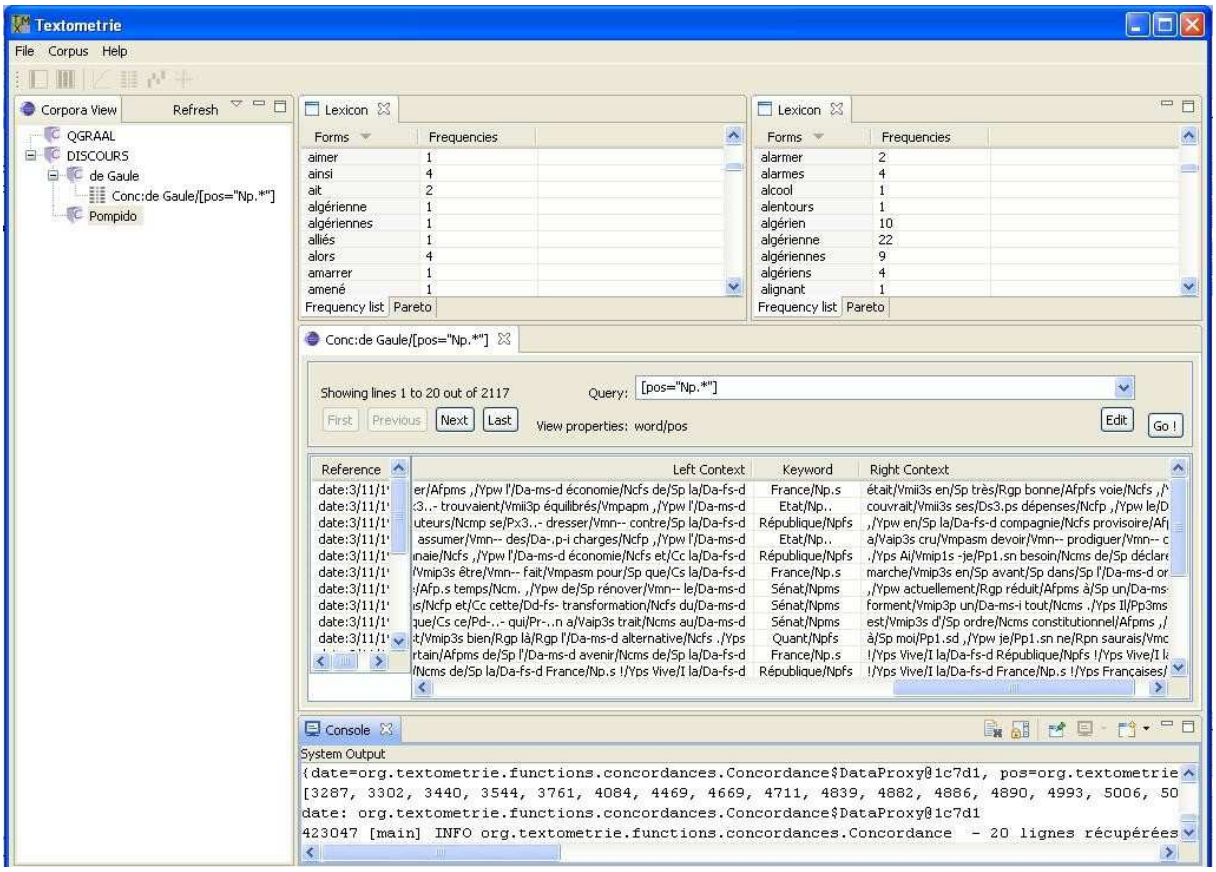

Figure 3 : exemple d'interface du prototype d'application locale RCP en environnement Windows. Le panneau latéral gauche donne accès, sous forme arborescente, aux corpus, sous-corpus, partitions, etc. c'est-à-dire tous les objets construits pendant la session de travail. Le panneau supérieur droit a été divisé en deux pour pouvoir comparer deux listes de vocabulaire. Le panneau inférieur droit présente une concordance détaillée dans la figure 4.

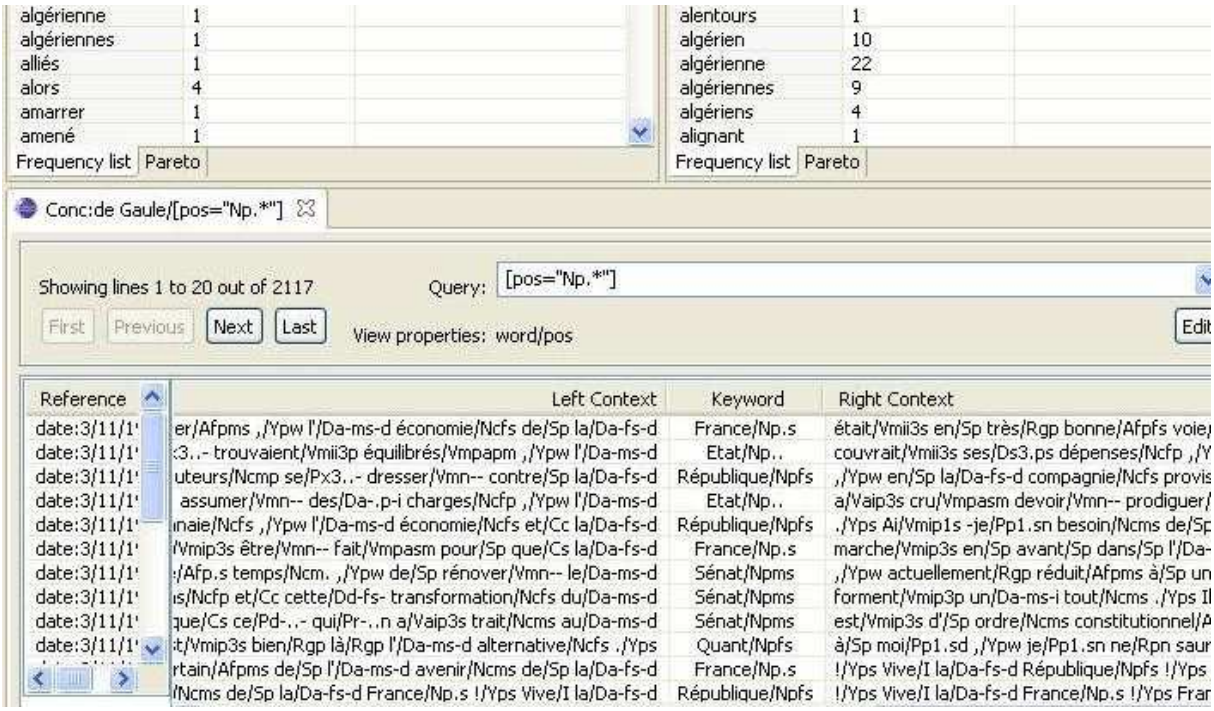

Figure 4 : détail de la concordance de la figure 3. On peut remarquer que dans cet exemple les occurrences sont affichées en associant leur forme et leur propriété morphosyntaxique, et que la référence de localisation de la ligne de concordance consiste en la date du discours. En effet, l'interface permet de définir dynamiquement les propriétés d'affichage comme la composition des références de localisation. Le champ « Query » contient l'expression algébrique du focus « [pos= "Np.*"] » qui s'interprète comme « toutes les occurrences dont la propriété « pos » commence par les caractères « Np », c'est à dire « tous les noms propres » dans le jeu d'étiquettes morphosyntaxiques de l'outil d'étiquetage utilisé pour ce corpus (fourni par Damon Mayaffre).

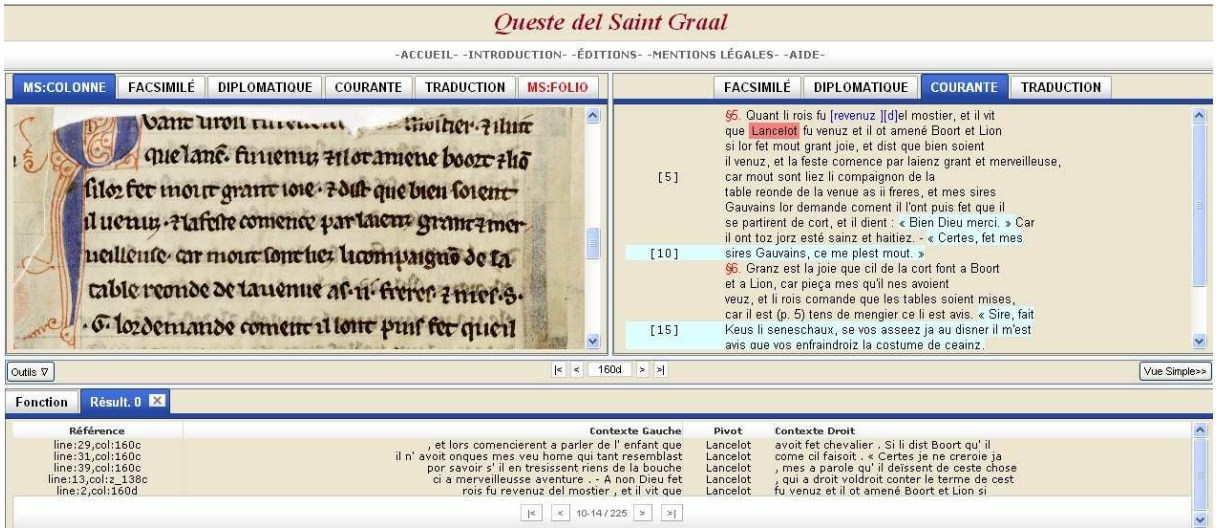

Figure 5 : exemple d'interface du prototype d'application web. Le panneau supérieur de présentation des éditions a été séparé en deux afin de pouvoir comparer l'image du manuscrit à la transcription de l'édition critique. Le panneau inférieur présente une concordance de la forme « Lancelot ». Un double clic sur la dernière ligne de cette page de concordance a provoqué l'affichage dans le panneau supérieur de la colonne « 160d » sous les deux facettes demandées, avec la mise en évidence de l'occurrence en rouge dans l'édition critique.

6. Conclusion

Plutôt qu'un logiciel de textométrie, le développement entrepris amorce donc la réalisation d'une pluralité de logiciels, adaptés à différentes communautés d'utilisateurs, tout en mutualisant et en optimisant l'investissement informatique. La plateforme TXM concerne tant les concepteurs et développeurs de logiciels (de par son versant "boîte à outils") que les utilisateurs d'outils d'analyse textuelle (via les applications réalisées en intégrant les composants de la boîte). Une attention particulière est portée sur la modularité et la généricité des composants : ainsi, un même composant, spécialisé par exemple pour la recherche de motifs, peut être mobilisé tant pour une application installée en local, sur un ordinateur donné, que pour une application en ligne, pour des usages mobiles ou collaboratifs. Le second choix informatique déterminant concerne l'ouverture du code. Ainsi, la plateforme n'est pas l'œuvre d'une personne, d'un laboratoire, d'un projet, mais l'objectif est que la communauté de la textométrie puisse librement se l'approprier et la faire évoluer. L'enjeu est tout à la fois celui de l'évaluabilité précise du code, de la pérennité et de la dynamique d'évolution des implémentations. C'est aussi l'enjeu de la mutualisation des développements sans sacrifier la

richesse d'une diversité de réalisations logicielles reflétant les multiples approches de l'analyse statistique des données textuelles.

Cette communication a été préparée dans le cadre du projet Textométrie ANR-06-CORP-029 ; elle a bénéficié d'une réflexion collective dépassant le cercle des auteurs ayant rédigé ces lignes.

Références

Christ O. (1994). A Modular and Flexible Architecture for an Integrated Corpus Query System. In *Proc. of COMPLEX'94 (3rd Conf. on Computational Lexicography and Text Research)*, pp. 23-32.

Fénelon J.-P. (1981). *Qu'est-ce que l'Analyse des Données*. Lefonen, Paris.

Heiden, S. (2006), - Un modèle de données pour la textométrie : contribution à une interopérabilité entre outils, *Actes des 8es Journées internationales d'Analyse Statistique des Données Textuelles (JADT'06) "Archives, Bases, Corpus"*, 19 - 21 Avril 2006, vol 1, p. 747-487, Jean-Marie Viprey et al., Presses Universitaires de Franche-Comté, Besançon, 2006.

Marchello-Nizia C. (*à paraître*). *Queste del saint Graal : Édition numérique interactive*, Lyon : ENS de Lyon.

Nancy I., Véronis J. (1994), - Multext (multilingual tools and corpora), In *Proceedings of the 15th CoLing*, pages 90-96, Kyoto, 1994.

Pincemin B. (2004). Lexicométrie sur corpus étiquetés. In Purnelle G. & al., editors, *Proc. of JADT 2004 (7es Journées internationales d'analyse statistique des données textuelles)*, pages 865-873.

Pincemin B. (2008) - Modélisation textométrique des textes, *Actes des 9es Journées internationales d'Analyse statistique des Données Textuelles (JADT 2008)*, Lyon, 12-14 mars 2008, Serge Heiden & Bénédicte Pincemin (éds), Lyon : Presses Universitaires de Lyon, vol. II, pp. 949-960.

Pincemin B., Heiden S., Lay M.-H., Leblanc J.-M., Viprey J.-M. (2010) – Fonctionnalités textométriques : proposition de typologie selon un point de vue utilisateur.

ISO/IEC 8859-1 (1998) - *8-bit single-byte coded graphic character sets, Part 1: Latin alphabet No. 1, April 15, 1998)*

R Development Core Team (2005) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

W3C Consortium (2004), Scalable Vector Graphics (SVG) format, Version 1.1, http://www.w3.org/TR/SVG11

TEI Consortium (2008) *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Version 1.1.0., 4th July. Lou Burnard & Syd Bauman eds., TEI Consortium. http://www.tei-c.org/Guidelines/P5.

Unicode Consortium (2006) *The Unicode Standard*, Version 5.0, 3rd November, Addison-Wesley Professional; 5th ed., http://unicode.org

DB XML : http://www.oracle.com/technology/products/berkeley-db/xml/index.html

Eclipse RCP : http://www.eclipse.org/home/categories/rcp.php

eXist-db : http://exist.sourceforge.net

Grails : http://grails.org

Groovy : http://groovy.codehaus.org

IMS CWB : http://cwb.sourceforge.net

JCP : http://jcp.org

Manatee : http://www.textforge.cz

OSGi : http://www.osgi.org

R : http://www.r-project.org

Sedna : http://modis.ispras.ru/sedna

Sourceforge : https://sourceforge.net

Spring : http://www.springsource.org

TEI : http://www.tei-c.org

TXM : http://www.textometrie.org

Xaira : http://www.oucs.ox.ac.uk/rts/xaira