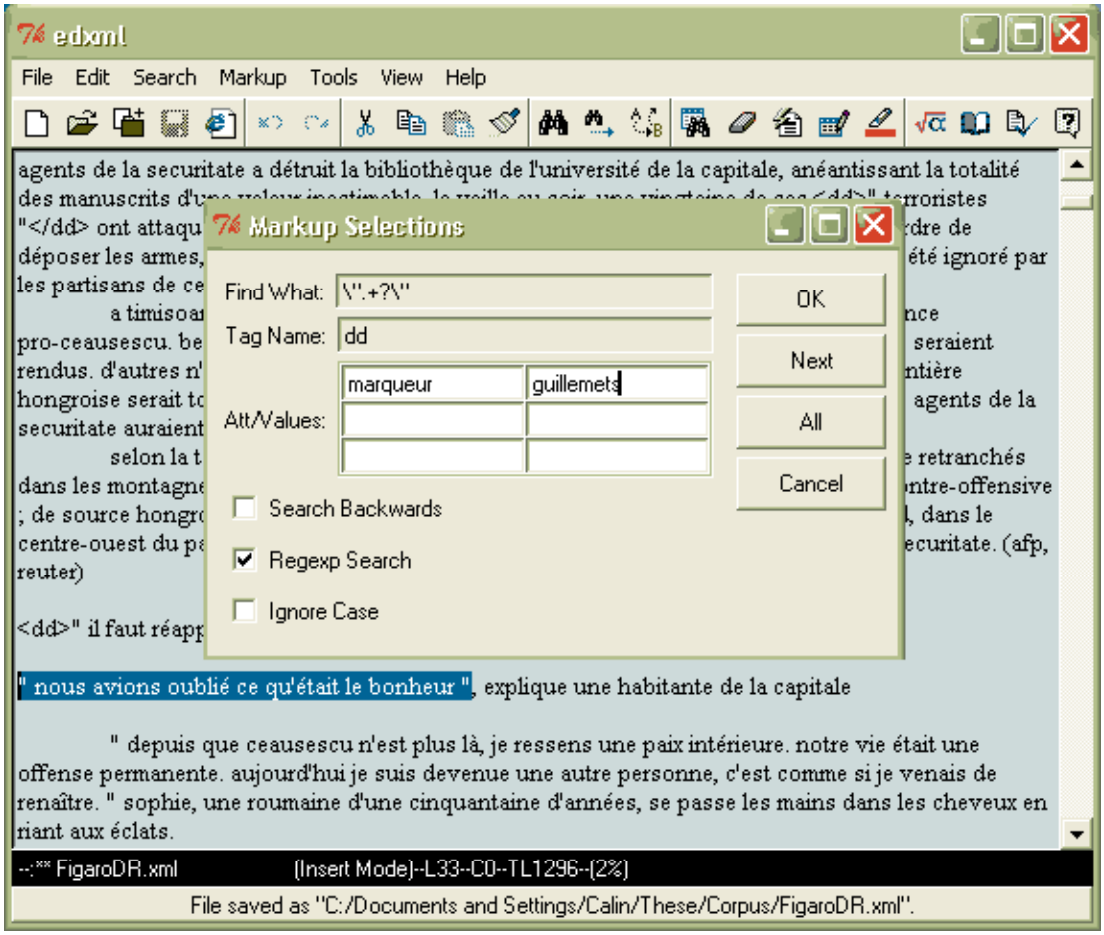
# UNIVERSITÉ PARIS III – SORBONNE NOUVELLE

## ÉCOLE NORMALE SUPÉRIEURE DE LYON


**documentation du module *edxml***

# 1 Préambule

Le projet ***edxml*** poursuit le développement d'une interface utilisateur standardisée permettant l'exploration de la proximité formelle entre des séquences textuelles, le placement assisté de marques dans un corpus, l'extraction, l'effacement et l'exportation des items marqués.




## 1. Balisage assisté d'un fichier en édition

Développé dans le cadre de l'Ecole Normale Supérieure de Lyon, le module ***edxml*** propose des fonctionnalités spécifiques pour la manipulation des fichiers de texte dans un tampon graphique, comme des outils de recherche-remplacement adaptés à l'édition des balises ***XML***. Il implémente des routines originelles réunies dans les bibliothèques ***Tk::TextTools.pm*** et ***Tk::MarkupTools.pm***, présentant la structure d'un éditeur de texte classique : une fenêtre de premier niveau, des menus et des boutons. Les opérations sur le fichier chargé peuvent être contrôlées depuis le menu

principal ou depuis les boutons, qui sont des raccourcis pour les fonctions déjà listées dans le menu.

L'interface est conforme au standard ***Windows***. Les messages de la barre d'état et les bulles d'information affichées temporairement au-dessus de certains widgets guident l'utilisateur durant les traitements. Une fonction de rappel est responsable de l'activation ou la désactivation des contrôles en fonction des événements intervenus dans le tampon d'édition. L'état désactivé d'un bouton ou d'une entrée de menu indique que l'option respective n'est pas disponible sur le moment et facilite une prise en main intuitive des fonctionnalités proposées. Il est possible de demander l'affichage de l'aide depuis la barre de commande du module ou depuis le menu contextuel du tampon d'édition.

Les expressions rationnelles sont utilisées par de nombreux programmes ***Unix-Linux*** comme **grep**, **sed**, **awk**, des éditeurs comme **emacs** et par certains shells.[1] L'outil ***sed*** permet, par exemple, de trouver toutes les instances d'un motif de recherche dans un fichier fourni en argument et les remplacer éventuellement avec une séquence de texte, comprenant inclusivement les références arrière.[2] Nous nous proposons de porter ces facilités dans un environnement graphique qui offre plusieurs modes de balisage assisté en fonction de la valeur des marques : automatique ou occurrence par occurrence. **edxml** comporte un des moteurs de recherche les plus performants à l'heure actuelle grâce à l'implémentation du langage ***Perl***[3] de programmation : un automate à états finis non déterministe, optimisé avec l'algorithme de **Boyer-Moore**.

Nous avons inclus dans les fonctionnalités de **edxml** la vérification des couples de caractères spéciaux. Le module recherche dans le tampon d'édition les caractères pairs spéciaux (***{[(<« »>)]}***) qui entourent le curseur, en essayant de dépister les erreurs d'alignement ou de clôture et en marquant avec du rouge les positions considérées comme défectueuses.

---

[1] Une **expression rationnelle** est une manière formelle de décrire un ensemble de chaînes sans devoir toutes les énumérer. Les **expressions rationnelles** peuvent être utilisées pour déterminer si une chaîne correspond à un motif donné. Il existe de nombreux modificateurs utilisables pour réaliser l'indifférentiation des majuscules et des minuscules en recherchant des chaînes de caractères ou des positions précises dans les chaînes de caractères.

[2] Ce qui permet de placer automatiquement des marques autour des séquences couvertes par le motif de recherche.

[3] Le **Practical Extraction and Report Language** est à l'origine un langage d'intégration pour le système d'exploitation ***Unix***, particulièrement adapté au traitement de textes.

**edxml** est un logiciel libre. Nous convions l'utilisateur à bien vouloir se rapporter aux termes de la licence concernant le mode d'utilisation ou de redistribution du code informatique en l'**ABSENCE DE TOUTE GARANTIE** de la part de l'auteur pour le bon fonctionnement des modules compilés.

# 2 Présentation du support CD-ROM

Le module *edxml* a été compilé pour le système d'exploitation ***Windows*** avec le compilateur ***Perl2exe*** en version d'évaluation (7-01-win32), disponible pour le téléchargement sur le site http://www.indigostar.com. L'interface d'installation du module a été créée avec la version *2003.2.0* d'évaluation de l'utilitaire ***CreateInstall,*** disponible, à l'heure de la rédaction de cette étude, à l'adresse Internet http://www.gentee.com. L'icône du module a été créée en utilisant la version *6.2.7* d'évaluation de l'utilitaire ***IconForge***, disponible sur le site http://www.cursorarts.com.

Le logiciel sur le support CD-ROM a été développé dans le cadre de l'Ecole Normale Supérieure de Lyon. Il est distribué sous le contrat ***GNU GENERAL PUBLIC LICENSE***, qui prévoit expressément la liberté de distribuer des copies de logiciels libres, de modifier les programmes ou d'en utiliser des éléments dans de nouveaux logiciels, en transmettant aux bénéficiaires tous les droits stipulés dans la **Licence Publique Générale**.

Le module *edxml* étant présenté comme un logiciel de recherche, il reste dans la propriété intellectuelle de l'auteur de cette étude sans cependant faire objet d'un brevet. Il est distribué dans l'espoir qu'il sera utile, mais **SANS AUCUNE GARANTIE** pour le bon fonctionnement des programmes assemblés et **SANS LA GARANTIE IMPLICITE D'ADEQUATION A UN USAGE PARTICULIER**. Un exemplaire de la Licence Publique Générale ***GNU*** est fourni avec chaque copie de l'exécutable, en support électronique.

# 3 Installation du module *edxml*

Pour installer les composantes du logiciel, il faut télécharger l'archive ***edxml*** disponible sur la page *HTML* de distribution de ce manuel. L'archive contient le code source complet avec toutes les ressources et un exécutable **win32** associé à une interface d'installation. Vous pouvez également disposer du CD-ROM fournissant un support de développement analogue.

## 3.1 Installation de l'exécutable win32

Selon que vous disposez du CD ou vous avez téléchargé la source et l'exécutable ***edxml***, vous devrez parcourir les mêmes étapes d'installation/désinstallation.

Pour installer **edxml** :

1. Dans l'**Explorateur Windows**, double-cliquez sur le programme d'installation ***Setup*** situé dans le répertoire ***Install*** du CD-ROM ou du dossier archivé ***edxml***. L'installation suit une procédure standard sous **Windows**. Suivez les instructions qui s'affichent à l'écran.

Pour désinstaller **edxml** :

1. Cliquez sur le bouton ***Démarrer*** de **Windows**, pointez sur **Paramètres**, puis cliquez sur **Panneau de configuration**.
2. Double-cliquez sur l'icône ***Ajout/Suppression de programmes***.
3. Cliquez sous l'onglet ***Installation/Désinstallation*** sur **edxml**, puis sur **Ajouter/Supprimer**.

## 3.2 Compilation du module *edxml* à partir des sources

Nous partirons, dans cette section, du présupposé que l'utilisateur est un adepte de la programmation en **Perl**. Le cas échéant, toutes les précisions nécessaires pourront être demandées auprès de l'auteur du logiciel. Pour compiler **edxml** à partir des sources, il faut avoir installé au préalable **Perl** et **PerlTk**, avec les bibliothèques requises par le module main ***edxml.pl***. En voici la liste alphabétique :

**BerkeleyDB, DB_File.pm, Expat, Storable.pm, Tk-MarkupTools.pm, Tk-Splash.pm, Tk-TextTools.pm, Tk-XMLViewer.pm, Unicode::String.pm, XML::Parser.pm**.

L'utilisateur trouvera des versions ***Windows*** précompilées de ces bibliothèques dans le répertoire ***Libraries*** du CD-ROM ou de l'archive ***edxml*** téléchargée. Les modules ***PM***, excepté **Tk-TextTools** et **Tk-MarkupTools**, correspondent à des architectures différentes de **ActivePerl**. Pour les installer, il est nécessaire de parcourir ces trois étapes :

1. Récupérer le contenu des archives et vérifier les compatibilités.
2. Mettre à jour **ActivePerl** (passage aux versions ***5.6.1*** et ***5.8.0***).
3. Installer les modules utilisant **ppm2** (dans une fenêtre ***DOS***, taper : **ppm install module.pm**).

Pour installer les deux modules ***PM*** développés dans le cadre du projet ***edxml***, il faut utiliser **Nmake**, fourni dans le répertoire ***Shareware*** du CD-ROM ou de l'archive téléchargée. Il propose des fonctionnalités analogues à celles de l'outil ***make*** sous

**Unix-Linux**. Après l'expansion des archives, lancer sous le **shell** de votre système les commandes suivantes :

1. **perl Makefile.PL** ; **Nmake test** ; **Nmake install**.

L'installation de la bibliothèque ***BerkeleyDB*** requiert un compilateur adapté à votre système d'exploitation, protégé probablement par une licence commerciale.

# 4 Les entrées de menu : mode d'emploi

## *4.1* Le menu *File*

L'entrée ***New*** commande le chargement d'un nouveau fichier en mode lecture-écriture. Lors de l'édition d'un document, l'indicateur textuel sur la barre d'état précise si la frappe s'effectue en mode ***Insertion*** ou en mode ***Refrappe***. Pour basculer entre le mode ***Insertion*** et le mode ***Refrappe***, presser la touche ***Inser*** du clavier. Par défaut, la frappe remplace la sélection. Un menu contextuel peut être affiché à tout moment en cliquant avec le bouton droit de la suris dans le tampon d'édition. Les options minimales d'édition (taille du texte, définition du caractère utilisé) peuvent être configurées depuis le menu ***Tools***.

L'entrée ***Open*** appèle une boîte de dialogue permettant la sélection et l'ouverture d'un fichier enregistré sur les disques locaux. Dans la boîte de dialogue, vérifier que le type de fichier approprié est sélectionné dans la zone ***Type de fichiers***. Pour faciliter la navigation et le tri des fichiers, deux filtres restrictifs sont proposés : celui qui affichera seulement les fichiers ***TXT*** et celui qui affichera seulement les fichiers ***XML***.

La commande ***Include*** append au point d'insertion le contenu d'un fichier sans effacer l'information ***Undo*** sur les changements intervenus dans le tampon graphique.

L'entrée ***Save*** déclenche l'enregistrement du texte édité, en appelant une boîte de dialogue si le nom du fichier ne figure pas encore dans la mémoire de l'ordinateur. ***Save As*** affiche un dialogue permettant le choix du nom du fichier à enregistrer et de son emplacement. Pour enregistrer rapidement un document, cliquer sur le bouton ***Enregistrer*** dans la barre des boutons. L'utilisateur doit spécifier explicitement l'extension du fichier dans la zone ***Nom de fichier***, le filtre utilisé ayant uniquement la fonction de tri des fichiers affichés dans la fenêtre d'exploration. L'interface manipule uniquement du texte simple et aucune conversion n'est effectuée au moment de l'enregistrement d'un fichier portant une extension autre que **TXT**.

La fonction d'exportation de la sélection primaire a été développée dans le but de faciliter l'exploitation des séquences textuelles marquées. L'entrée ***Export To*** commande l'enregistrement du texte sélectionné dans un fichier ouvert moyennant un dialogue. Il peut recevoir plusieurs contenus sans l'effacement des séquences exportées précédemment.

La commande ***Exit*** permet de quitter l'application. Un protocole de sortie déclenche la vérification des indicateurs de sauvegarde du tampon graphique avant la fermeture effective du programme.

## 4.2 Le menu *Edit*

Les fonctions d'édition appelées depuis le menu ***Edit*** correspondent au standard *Windows*. On peut sélectionner du texte en utilisant la souris ou le clavier. Pour revoir les associations d'événements graphiques prédéfinies, aller à la section ***§ 5***. Supposant que les commandes listées ci-dessus sont déjà familières à l'utilisateur, nous nous contenterons d'en donner un bref descriptif.

**Undo** : annule la dernière action.
**Redo** : refait pas à pas les dernières modifications du tampon d'écriture.
**Cut** : efface la sélection primaire de la fenêtre d'édition et la stocke dans le CLIPBOARD.
**Copy** : mémorise la sélection primaire de la fenêtre d'édition dans le CLIPBOARD.
**Paste** : insère le contenu du CLIPBOARD dans la fenêtre d'édition.
**Select All** : marque toutes les lignes du tampon graphique comme sélection primaire.
**Unselect All** : efface les zones sélectionnées dans la fenêtre d'édition.

## 4.3 Le menu *Search*

Nous avons développé les routines de recherche-remplacement de manière à adopter l'usage des expressions régulières dans la définition des phénomènes textuels. La possibilité de demander au module l'index des formes graphiques du fichier chargé dans le tampon d'édition facilite l'écriture des motifs complexes.

Si une recherche est lancée sur un fichier alors qu'il existe une sélection primaire dans l'application, la chaîne sélectionnée sera retrouvée dans le champ de saisie du motif. Si la sélection est effectuée ultérieurement, un clic avec le bouton ***3*** de la souris transforme la chaîne sélectionnée en une expression régulière avant de la transmettre à la fenêtre de commande de la recherche des motifs. Il devient facile, par la suite, de créer des motifs complexes en effectuant une sélection multiple dans l'index des formes graphiques et en la concaténant automatiquement. Les mêmes fonctionnalités seront retrouvées dans les fenêtres de premier niveau gérées depuis le menu ***Markup***.

La commande ***Find*** déclenche la recherche de la chaîne donnée en argument dans la fenêtre d'édition. Les options ***direction***, ***casse*** et ***mode*** sont contrôlables. Le mode ***regexp*** constitue un des arguments les plus importants de notre module. Les expressions régulières ***Perl*** sont très performantes et incluent les références en arrière. La recherche peut être globale.

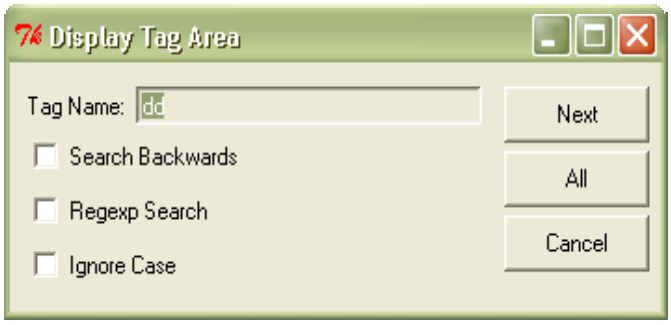
L'entrée ***Find Next*** commande une recherche ***forward*** (en avant) de la chaîne fournie en argument, s'arrêtant à la première occurrence du motif trouvée.

L'entrée ***Find Previous*** commande une recherche ***backward*** (en arrière) de la chaîne fournie en argument, s'arrêtant à la première occurrence du motif trouvée.

La commande ***Replace*** déclenche la recherche de la chaîne fournie dans la fenêtre d'édition et remplace ses instances avec une autre chaîne de caractères. La recherche est paramétrable (les arguments sont identiques à ceux de la commande ***Find***). La recherche-remplacement des motifs peut être globale.

## 4.4 Le menu *Markup*

Ce menu réfère aux fonctions responsables de la manipulation des balises ***XML*** standard. L'utilisateur est convié à prendre connaissance du fait que leur représentation formelle au moyen des expressions régulières (**<[^>]+>**) ne représente pas une solution infaillible. Le module n'aura pas le comportement escompté en la présence de balises scindées sur plusieurs lignes. La plupart des routines suivantes présentent des boucles distinctes correspondant à deux modes d'édition : le mode assisté (***Safe***) et le mode non assisté (***Unsafe***), qui désactive le correcteur syntaxique. Si l'application le requiert, l'utilisateur peut remettre à jour l'état des indicateurs ***XML*** et indenter éventuellement les lignes, en pressant la touche ***F5***. Voir § ***7.2*** pour les restrictions dues à la taille des fichiers.

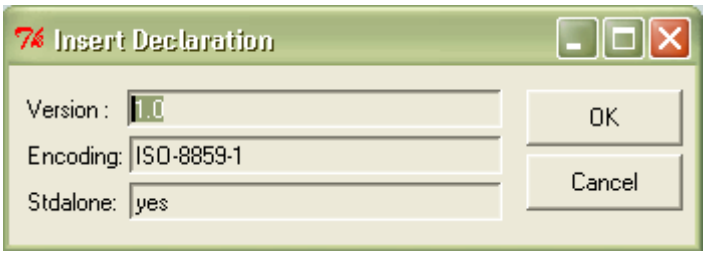



**2. Sélection des champs décrits par une balise**

L'entrée ***Display Tag Area*** commande la recherche des séquences de texte marquées avec le couple de balises d'ouverture et fermeture proposé en argument. La recherche peut être globale (voir la copie d'écran ***2***).

La commande ***Delete Tag Area*** recherche les séquences de texte marquées avec le couple de balises d'ouverture et fermeture proposé en argument et les efface. La recherche peut être globale.

La commande ***Remove Markups*** efface les balises standard du fichier édité.



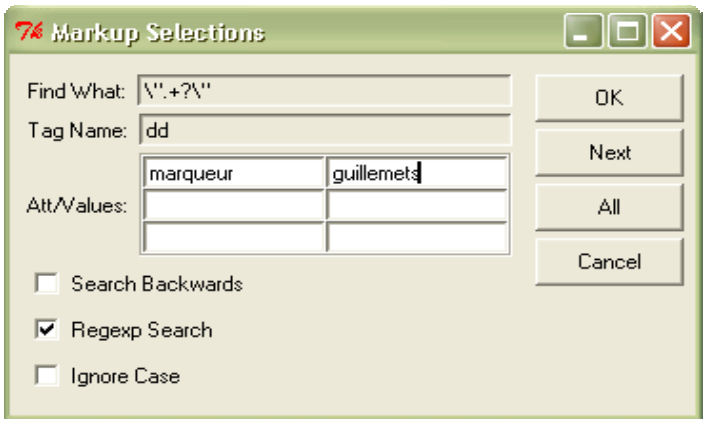

## 3. Edition assistée de la DTD

L'entrée ***XML Declaration*** déclenche une vérification de la syntaxe (en mode assisté) et autorise le placement de la déclaration ***XML*** au point d'insertion du tampon graphique (voir la copie d'écran ***3***). Cette balise doit constituer le premier élément du document en édition et doit être unique. La commande ***Document Type*** déclenche la vérification syntaxique et place la balise du type du document au point d'insertion du tampon graphique. Cet élément doit occuper la deuxième position dans la structure logique du fichier et doit être unique. Si les indicateurs ***XML*** ne confirment pas ces statuts, les instruments d'édition seront temporairement désactivés.

L'entrée ***Insert Comment*** insère un commentaire ***XML*** dans le fichier édité. En mode assisté, le module vérifie la présence d'éventuels caractères spéciaux (à risque) dans la chaîne saisie par l'utilisateur.

L'entrée ***Write Tag*** commande l'affichage d'une boîte de dialogue requerrant un nom de balise et des attributs éventuels. Lorsque le bouton ***Ok*** est pressé, une balise d'ouverture, une balise vide ou une balise de fermeture est placée au point d'insertion du tampon graphique. En mode assisté, l'insertion d'une balise déclenche la vérification préalable de l'arborescence ***XML***. Toute modification du fichier en édition se répercute sur la structure mémorisant le niveau des éléments et leur état (ouverture ou fermeture).

La commande ***Insert Text*** insère une chaîne textuelle dans le fichier édité. Les caractères spéciaux sont directement traduits en entités avant l'écriture de la séquence dans le tampon graphique. En mode assisté, le module vérifie la position du point d'insertion dans l'arborescence ***XML***.

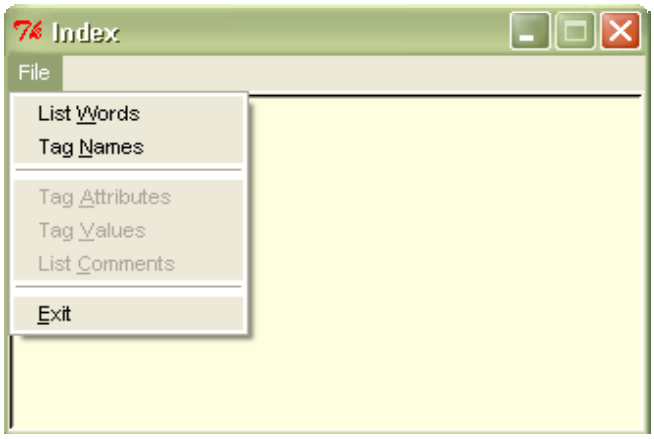



## 4. La boîte de commande de la recherche de motifs et d'écriture des balises

L'entrée ***Markup Selections*** permet l'affichage d'un dialogue requerrant la définition d'un motif de recherche avec les paramètres ***mode*** et ***casse***. La fonction procède au déroulement progressif du buffer, au fur et à mesure que les instances des motifs de recherche sont reconnus dans le texte. L'utilisateur a la possibilité de contrôler le marqueur de sélection durant le lancement d'un traitement impliquant des motifs de recherche qui laissent une marge d'erreur non négligeable. Lorsque les choix sont confirmés avec le bouton ***Ok***, la sélection est encadrée par des balises ***XML*** d'ouverture et de fermeture.

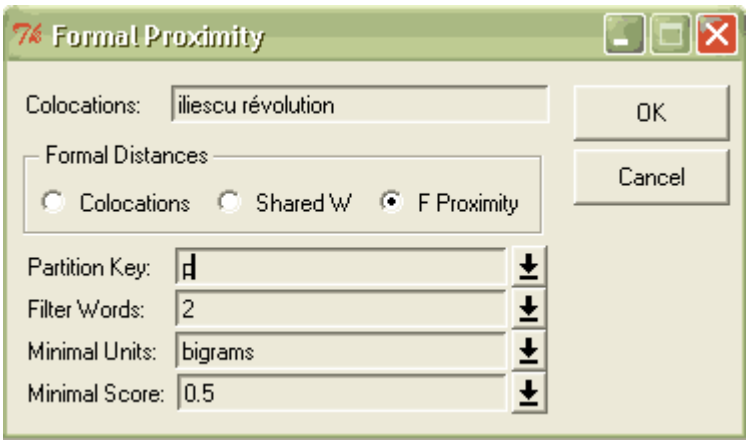
**Remarque** : Le balayage à plusieurs reprises du tampon d'édition permet une adaptation permanente de la stratégie de balisage aux caractéristiques de chaque corpus. Des motifs de recherche complexes peuvent être composés à partir d'une sélection multiple dans la liste des formes lexicales.

## 4.5 Le menu *Tools*




**5. La fenêtre d'affichage de l'index avec son menu**

L'entrée ***Show Index*** est responsable de la création d'une fenêtre de premier niveau contenant un widget ***liste*** (voir la copie d'écran ***5***). L'utilisateur peut demander l'extraction des formes lexicales, des noms de balises, des attributs, valeurs et commentaires du fichier en édition. Le balayage du buffer fait appel à des mécanismes distincts, selon que le fichier présente l'extension ***XML*** (il sera alors filtré au moyen du parser ***Expat***) ou non. Les entrées du menu seront activées en fonction de l'extension du fichier chargé. Voir aussi § ***7.2*** pour les limites du traitement proposé. Nous avons associé le widget ***liste*** à des fonctions du module ***XML-RegExp.pm***. Elles autorisent la construction d'un motif de recherche du type ***ou... ou***, à partir d'un index de mots.




**6. La fenêtre de commande *Formal Proximity*, avec l'option de calcul des distances en bigrammes activée**

L'activation de l'entrée ***Formal Proximity*** lance une fenêtre de contrôle des routines qui calculent la distance formelle entre les chaînes textuelles. Ce module de commande appèle trois fonctions principales correspondant aux traitements suivants : recherche de **collocations**, représentation schématique de la distribution des formes lexicales dans le corpus traité (***Shared Words***) et **calcul des énoncés proches**. Pour le descriptif des algorithmes, voir **MOSUT** (2003). Détail des traitements :

L'**indexation** du corpus précède obligatoirement la recherche des **collocations** mais on ne parcourt qu'une fois cette étape lors des traitements successifs. La fenêtre de commande ayant recueilli un certain nombre de formes lexicales, on extrait du tampon d'édition les énoncés qui manifestent des occurrences de ces mots. La routine impliquée ne fait pas appel aux expressions régulières mais à la lecture matricielle de la structure issue de l'**indexation**. Les clefs du hachage sont consultées et l'on incrémente chaque apparition des indices d'énoncés dans les tableaux ; sont retenus les indices incrémentés un nombre de fois égal à la longueur de la liste d'entrées.

**Remarque** : Les **collocations** agissent comme un filtre restrictif, en éliminant du calcul des proximités les énoncés qui ne manifestent pas la coprésence des formes lexicales proposées en argument.

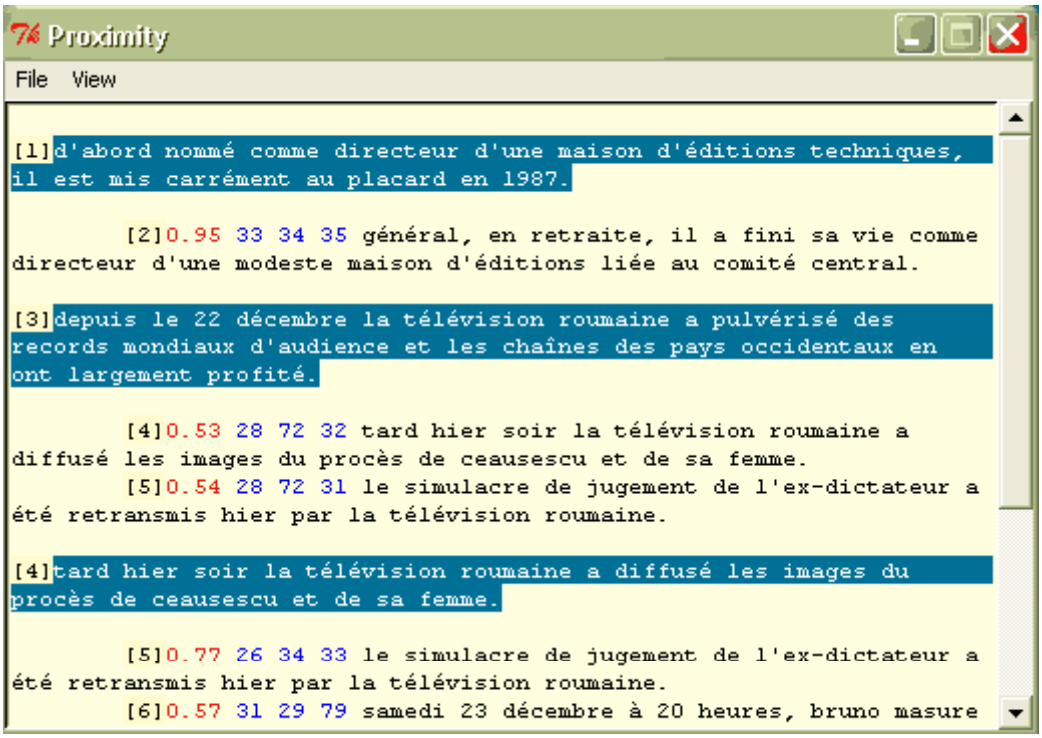
L'option ***Shared Words*** commande une lecture matricielle de l'index d'adressage des formes lexicales pour en extraire les couples d'adresses (indices d'énoncés ) manifestant la présence d'un nombre de **formes lexicales** communes. On peut exprimer les phénomènes de **proximité** en valeurs absolues : deux **énoncés** sont **proches** parce qu'ils partagent un nombre donné de formes lexicales distinctes ou certaines formes lexicales prédéfinies. L'index de l'application permet à l'utilisateur de proposer les **mots** qui manifestent des occurrences dans les énoncés comparés.

**Remarque** : Les **mots partagés** agissent comme un filtre restrictif, en éliminant du calcul des proximités les énoncés qui ne manifestent pas le nombre de formes lexicales communes proposé en argument.

Les fonctions activées avec l'option ***Formal Proximity*** lisent l'index des distances entre les formes lexicales d'un couple d'énoncés et réalisent des comparaison entre des ensembles de **mots** graphiques ou de **bigrammes** selon l'**algorithme d'Adamson et Boreham**. Si l'on a choisi les **mots** comme unités minimales, la distance entre énoncés est calculée en fonction des formes lexicales communes et des formes spécifiques à chaque énoncé, après l'élimination des doublons. Si l'option correspondant aux **bigrammes** a été activée, le nombre de **bigrammes** communs et distincts pour deux énoncés est calculé par addition et soustraction, après l'échelonnement des distances entre les formes lexicales. En effet, du fait que le **bigramme** perd ses traits discriminants au niveau de l'énoncé, les premiers décomptes peuvent être faits uniquement au niveau des formes lexicales. Le filtrage des sorties est effectué à partir d'un seuil fixé empiriquement au début des traitements

et visant à sélectionner parmi un grand nombre de résultats, ceux pour lesquels les valeurs d'un indice numérique dépassent ce seuil.

**Remarque** : L'algorithme d'**ADAMSON et BOREHAM** (1974) met en oeuvre une comparaison entre les éléments minimaux d'une chaîne lexicale, en donnant un rapport sur la présence et l'absence de ceux-ci.




## 7. Présentation des résultats du calcul de la distance formelle entre les énoncés

Le calcul des distances formelles à partir des **bigrammes** s'effectue en deux étapes, dont la première consiste en l'établissement des meilleurs **scores de proximité** pour chaque forme de l'un des énoncés termes de la comparaison. La seconde étape concerne la réduction des télescopages : deux ou plusieurs formes de l'énoncé ***A*** peuvent présenter des affinités pour la même forme de l'énoncé ***B***. Il faut favoriser dans ce cas une des formes de l'énoncé ***A*** et réorienter les autres en fonction des places disponibles ou les conserver simplement pour le calcul de la **différence symétrique**, si toutes les places ont été déjà occupées avec des **scores de proximité** supérieurs. Le calcul de l'**indice *Dice*** est demandé pour chaque couple de formes lexicales comparées. La **différence symétrique** s'obtient en définissant une somme totale de bigrammes pour chaque énoncé, égale au nombre de bigrammes uniques rencontrés dans la forme *1* plus le nombre de bigrammes uniques rencontrés dans la forme *n* de l'énoncé et en soustrayant de cette somme les bigrammes partagés issus

du tri de l'index. Le retour des valeurs permet d'évaluer la **proximité** entre les énoncés (voir la copie d'écran *7*).

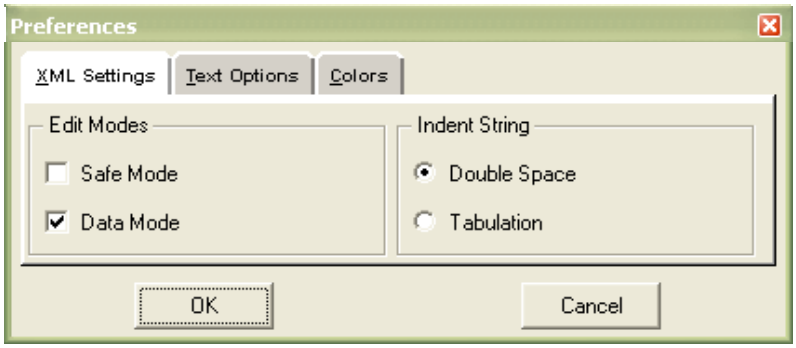
**Remarque** : On peut spécifier explicitement les entrées de l'index, simplifiant ainsi le protocole de comparaison. De même, une clef de partition fournie évite au programme beaucoup d'itérations. Les **formes graphiques** constituent des unités facilement dénombrables, mais leur variation flexionnelle est problématique. Les **bigrammes** ne peuvent être comptés qu'après l'examen des affinités entre les formes lexicales et le calcul de la meilleure superposition pour un couple d'énoncés, mais ils donnent une mesure plus pertinente de la **proximité**.

L'entrée ***XML Parser*** est responsable de la création d'une fenêtre de premier niveau qui affiche les résultats du parsage du buffer ou d'un fichier ***XML*** externe. Les routines impliquées interfacent le parser ***Expat*** en associant des fonctions de rappel aux handles d'événements ***XML*** intervenus lors du filtrage des chaînes textuelles. La fenêtre de présentation de l'arborescence ***XML*** comporte un affichage à couleurs et des zones actives qui autorisent l'extension ou la contraction des nœuds visités. Un menu contextuel associé au ***parser*** permet la visualisation de l'en-tête des fichiers.

## 4.6 Le menu *View*

La commande ***Goto Line*** place le curseur au début de la ligne du fichier demandée par l'utilisateur.

L'entrée ***Wrap*** contrôle la présentation du texte dans le tampon d'édition. L'utilisateur peut demander l'affichage des lignes avec coupure sur le mot, sur le caractère ou sans coupure.




## 8. Configuration du module *edxml* pour le balisage assisté d'un fichier

L'entrée ***Preferences*** lance la fenêtre de configuration minimale de l'application. Il s'agit des paramètres d'affichage du texte et des modes d'édition ***Safe*** et ***Unsafe***. Certaines entrées des menus seront activées ou désactivées en fonction du mode

d'édition sélectionné. Le module ***edxml*** comporte un fichier de configuration modifié chaque fois que les paramètres d'édition changent.

## 4.7 Le menu *Help*

Nous nous contenterons d'énumérer les entrées de ce menu et de préciser leurs affectations.

**Help** : requiert le fichier d'aide du module (rédigé en anglais). Du fait que la bibliothèque ***Tk*** ne comporte pas une prise en charge multilingue, nous avons accepté l'anglais comme langue unique de développement.
**License** : présente la licence du module dans une fenêtre de premier niveau.
**About edxml** : affiche la fenêtre de présentation du module.

# 5 Traitement des événements graphiques dans l'application

Il est possible d'accomplir rapidement certaines tâches réalisées fréquemment en utilisant les touches de raccourci ou la souris. Nous proposons dans cette section du manuel une aide-mémoire sur les raccourcis courants.

## 5.1 Evénements associés à la souris

Un clic avec le bouton ***1*** de la souris dans le buffer positionne le curseur devant le caractère visé, transmet le focus sur la fenêtre d'édition et efface toute sélection dans celle-ci. On peut définir une sélection primaire en déplaçant la souris avec le bouton ***1*** enfoncé depuis le point d'insertion jusqu'au caractère au-dessus duquel on relâche la pression. Deux clics avec le bouton ***1*** de la souris sélectionnent le mot visé et positionnent le point d'insertion au début du mot. Déplacer la souris dans le texte après un double clic, en maintenant le bouton ***1*** pressé, définit une sélection regroupant des mots entiers.

En cliquant trois fois avec le bouton ***1*** de la souris on sélectionne la ligne au-dessus de laquelle se trouve la souris et l'on positionne le point d'insertion au début de la ligne. Déplacer la souris dans le texte, après un triple clic, en maintenant le bouton ***1*** pressé, définit une sélection regroupant des lignes entières. Le fait de cliquer ave le bouton ***1*** de la souris avec la touche ***CTRL*** pressée repositionnera le curseur dans le texte sans affecter la sélection. Le clic avec le bouton ***3*** de la souris dans les fenêtres de commande crée une expression régulière depuis la sélection primaire définie dans l'application avant de la transmettre au widget ***entry*** de le fenêtre.

Lorsque des caractères sont saisis, ils sont insérés dans le tampon d'édition au point d'insertion, qui peut correspondre à la position du curseur. Si la souris est déplacée au-dehors de la fenêtre principale avec le bouton ***1*** enfoncé, le buffer déroulera le texte pour le rendre visible au cas où il en resterait outre la zone couverte par l'écran.

## 5.2 Evénements associés au clavier

Les touches ***GAUCHE*** et ***DROITE*** déplacent le point d'insertion d'un caractère à gauche ou à droite ; elles enlèvent également les marqueurs de sélection présents dans le texte. Si la touche ***GAUCHE*** ou ***DROITE*** est pressée avec la clef ***SHIFT***, le point d'insertion se déplace et la sélection est modifiée pour inclure le nouveau caractère. **CTRL**+**GAUCHE** et **CTRL**+**DROITE** déplacent le point d'insertion par caractères, tandis que **CTRL**+**SHIFT**+**GAUCHE** et **CTRL**+**SHIFT**+**DROITE** déplacent le point d'insertion par mots en agrandissant la sélection.

Les touches ***HAUT*** et ***BAS*** déplacent le point d'insertion avec une ligne vers le haut ou vers le bas et enlèvent la sélection du buffer. Si la touche ***HAUT*** ou ***BAS*** est pressée avec la clef ***SHIFT***, le point d'insertion se déplace et la sélection est modifiée pour inclure le nouveau caractère. **CTRL**+**HAUT** et **CTRL**+**BAS** déplacent le point d'insertion par paragraphes (lignes contenant des caractères de mots séparées par des lignes vides) tandis que **CTRL**+**SHIFT**+**HAUT** et **CTRL**+**SHIFT**+**BAS** déplacent le point d'insertion par paragraphes et agrandissent la sélection.

Les touches ***SUIVANT*** et ***PRECEDENT*** déplacent le point d'insertion en avant dans le texte ou en arrière du contenu d'une fenêtre d'édition et effacent les marqueurs de sélection. Si la clef ***SHIFT*** est pressée avec **SUIVANT** ou **PRECEDENT**, la sélection est modifiée pour inclure le nouveau caractère. **CTRL**+**SUIVANT** et **CTRL**+**PRECEDENT** déroulent le texte d'une page sans déplacer le point d'insertion ou affecter la sélection.

La touche ***DEBUT*** déplace le curseur au début de la ligne active et efface les marqueurs de sélection. La combinaison de touches ***SHIFT***+***DEBUT*** déplace le curseur au début de la ligne active et y repositionne le marqueur de début de sélection. La touche ***FIN*** déplace le point d'insertion à la fin de la ligne active et enlève tout marqueur de sélection du buffer. **SHIFT**+**END** déplace le curseur à la fin de la ligne active et y repositionne le marqueur de fin de sélection. La combinaison de touches ***CTRL***+***DEBUT*** déplace le point d'insertion au début du texte et efface les marqueurs de sélection. **CTRL**+**SHIFT**+**DEBUT** déplace le curseur au début du texte et repositionne le marqueur de début de sélection à ce point. **CTRL**+**FIN** déplace le point d'insertion à la fin du texte tout en effaçant les marqueurs de sélection. **CTRL**+**SHIFT**+**FIN** déplace le curseur à la fin du texte et y replace le marqueur de fin de sélection.

L'utilisateur découvrira des interférences entre les associations prédéfinies dans la hiérarchie de classes ***PerlTk*** (se conformant au standard ***Unix***) et celles déduites des nouvelles routines, malgré notre effort de respecter les habitudes formées sous ***Windows*** :

| | |
|---|---|
| **CTRL+N** | Ouverture d'un nouveau fichier |
| **CTRL+S** | Sauvegarde du fichier édité |
| **CTRL+E** | Exportation de la sélection primaire vers le fichier choisi, sans effacement des contenus antérieurs |
| **CTRL+/** | Sélection du contenu entier du buffer |
| **CTRL+** | Effacement du marqueur de sélection du buffer |
| **F5** | Mise à jour des indicateurs ***XML*** et indentation des lignes |
| **F6** | Calcul de la proximité formelle entre les chaînes textuelles |
| **F7** | Indexation du fichier édité |
| **F8** | Parsage du document édité ou d'un fichier ***XML*** externe |
| **F1** | Copie de la sélection primaire du tampon d'édition vers le CLIPBOARD |
| **F2** ou **CTRL+X** | Copie de la sélection primaire du tampon d'édition vers le CLIPBOARD et effacement du texte sélectionné |
| **F3** ou **CTRL+V** | Insertion des contenus du CLIPBOARD dans le tampon d'édition |
| **CTRL+F** | Recherche dans le buffer |
| **CTRL+R** | Recherche-replacement dans le buffer |
| **CTRL+M** | Placement assisté des balises ***XML*** dans le fichier édité |
| **CTRL+T** | Ecriture d'une balise ***XML*** |
| **CTRL+H** | Affichage de l'aide |
| **SUPPR** | Effacement de la sélection, si présente dans l'application. S'il n'y a pas du texte sélectionné, cette touche efface le caractère à droite du point d'insertion. |
| **RETOUR** | Effacement de la sélection, si présente. S'il n'y a pas du texte sélectionné, cette touche efface le |

| | |
|---|---|
| | caractère à gauche du point d'insertion. |
| **CTRL+D** | Effacement du caractère à droite du point d'insertion |
| **CTRL+K** | Effacement de tous les caractères depuis le point d'insertion jusqu'à la fin de la ligne. Si le curseur se trouve déjà à la fin de la ligne, cette combinaison de touches efface le caractère ***nouvelle ligne***. |
| **CTRL+O** | Ouverture d'une nouvelle ligne avec insertion d'un caractère ***nouvelle ligne*** devant le point d'insertion |

# 6 Précision sur les notions théoriques

Tout traitement qui implique l'utilisation des **marques textuelles** dans les définitions formelles doit être précédé d'une étape de vérification et de renforcement de ces **marques**. Dans **edxml**, une **ligne** est traitée comme une unité d'informations distincte. Ainsi, la révision du corpus en vue du traitement des **énoncés proches** poursuivra la réduction des retours à la ligne accidentels et de la ponctuation non significative. L'interface que nous proposons permet de réviser les coupures de ligne ainsi que de l'usage convenu de la ponctuation forte. En effet, les lignes générées par un **O.C.R**., par exemple, risquent de ne pas respecter les conventions graphiques.

L'**indexation** d'un corpus suppose deux étapes proprement dites, intrinsèquement liées. Il s'agit de la **délimitation des unités minimales** et de l'**adressage** de ces unités, avec la création d'un modèle de structure simplifiée du corpus. Cela permet ultérieurement le rassemblement immédiat des formes lexicales et la récupération du **contexte d'extraction**. On comprend par **unités minimales** les unités que l'on ne décompose pas en unités plus petites, caractéristiques pour un certain niveau de traitement (**énoncés**, **formes lexicales, bigrammes**).

Nous opérons, à la manière des logiciels statistiques connus, la distinction entre les **caractères délimiteurs** et les **caractères non-délimiteurs** sur l'ensemble des caractères qui entrent dans la composition du texte. Nous procédons à la segmentation automatique du texte en **occurrences** (suite de caractères non-délimiteurs bornée à ses extrémités par des caractères délimiteurs). On distingue, parmi les caractères **délimiteurs**, les **délimiteurs d'énoncés** et les caractères **délimiteurs de formes lexicales**. La ponctuation forte (**le point**, **le point d'exclamation**, **le point d'interrogation**) et le **retour à la ligne** sont **délimiteurs d'énoncés**.

On définit un **énoncé** comme une chaîne de caractères comprise entre deux **séparateurs d'énoncé**. **L'énoncé** est le contexte minimal fourni lors de la recherche de **collocations**. La notion de **collocation** définit une présence simultanée, mais non forcément contiguë de plusieurs **formes lexicales** données dans un **énoncé**. Toute suite de caractères de mot est considérée comme une occurrence de **forme lexicale**. On considère comme des caractères de **mot** tout alphanumérique, caractères accentués inclus, plus le sous-tiret (_), en précisant que ce choix est la définition en termes ***Perl*** du caractère de mot. Sont inclus dans les caractères **délimiteurs de mots** le tiret simple et l'apostrophe. Un **bigramme** est une suite de deux lettres consécutives.

Des **partitions** peuvent être définies au moyen de balises du type ***<identificateur chaîne = « valeur »>***. On rencontre dans un texte des **délimitations** inhérentes telles que les délimitations chronologiques, les délimitations indiquant la séquence des parties, etc. Le corpus peut comporter des **balises** indiquant ces délimitations logiques sous une forme codée, non ambiguë, compréhensible par la machine, les divers regroupements des **énoncés proches** présentant un réel intérêt pour les approches contrastives des corpus. D'un point de vue pragmatique, cela évite un nombre important d'opérations de calcul, réduisant ainsi les temps de traitement.

La structure logique que l'on met en place avec la routine d'**indexation** est la suivante : **CORPUS = LIGNES→ENONCES (calcul, adressage)→FORMES LEXICALES (calcul, adressage)→BIGRAMMES (calcul, adressage)**. Une structure de données associe les **clefs de partition** et les **adresses** des **formes lexicales** dans le tampon d'édition, ce qui permet l'attribution d'une **occurrence** précise à une des **partitions** indexées. C'est en fonction de ces adresses que sont orientées les **comparaisons de proximité**. Deux **énoncés** sont considérés comme **proches** s'ils partagent un certain nombre d'**unités atomiques**, avec lesquelles on calcule un **score**. Associé à un **seuil de proximité**, celui-ci permet le filtrage des sorties.

Nous procédons à un regroupement des **énoncés** qui manifestent le phénomène de **proximité formelle** pour les étudier dans leurs **contextes de manifestation** (comparer des distributions semblables constitue une pratique habituelle dans les analyses de contenu), afin d'en mesurer les mutations par rapport à un éventuel **énoncé-source** commun et réaliser une mise en parallèle des **attributs argumentatifs**. Les **énoncés proches** regroupés constituent un paradigme et cette disposition peut être considérée comme un concordancier des variantes d'un **énoncé-source** dans le corpus.

L'implémentation informatique de l'algorithme d'extraction des **énoncés proches** réalisée dans le cadre de notre recherche, se réclame de la méthode de recherche et d'extraction de modèles lexico-syntaxiques ***LSPE*** (***Lexicosyntactic Pattern***

***Extraction***) à travers la recherche d'arguments de structure partagés. La définition des relations « **proches** » ou « **lointaines** » entre les unités de discours (tels qu'acceptées dans notre analyse de contenu) est à la charge d'une formule mathématique qui élimine les appréciations subjectives. Le **score de proximité** est un paramètre essentiel pour la récupération des énoncés qui migrent, avec des changements plus ou moins importants de la structure, à l'intérieur d'un corpus.

L'indice ***Dice*** définit la distance entre deux ensembles d'**unités minimales** en fonction de l'**intersection** et de la **différence symétrique** des composants. Les facteurs de calcul de cet indice (le nombre d'**unités partagées** et le décompte des **unités spécifiques** pour chaque **énoncé** terme de la comparaison) sont précisés dans les sorties.

# 7 Limites de edxml

Cette section sera mise à jour au fur et à mesure que nous recevrons les remarques des lecteurs qui auront bien voulu tester notre module. En proposant un logiciel libre, nous n'avons pas porté une attention particulière au traitement des exceptions. Les options d'édition sont minimales, strictement fonctionnelles et n'incluent pas les facilités proposées par un logiciel commercial (filtres autorisant l'importation des divers formats de fichiers, correcteurs orthographiques, aide contextuelle avancée, etc.).

## 7.1 Durée des traitements

Un programme écrit en **Perl** est de cinq à dix fois plus lent que son équivalent écrit en **C**, tandis que la bibliothèque graphique ***Tk*** sollicite d'une manière pesante la mémoire virtuelle de l'ordinateur. Le nombre d'opérations impliquées dans le calcul des **énoncés proches** augmente de manière exponentielle avec la taille du fichier en édition. Il est nécessaire, par conséquent, d'avoir conscience des limites du module et de respecter quelques règles pratiques lorsqu'on envisage le traitement d'un corpus : travailler avec de petites tranches, placer des clefs de partition dans le texte, évaluer correctement les possibilités de l'ordinateur personnel.

L'utilisateur n'a pas la possibilité de contremander les traitements une fois lancés. Il peut, par contre, en suivre le déroulement moyennant les indices visuels offerts par le module (la barre de progression, les messages de la barre d'état, l'aspect particulier du curseur).

## 7.2 Dimension des fichiers

Les problèmes liés à la manipulation des gros fichiers imposent des restrictions concernant certaines fonctionnalités proposées. Il s'agit notamment des routines qui manipulent les éléments ***XML*** et des routines de calcul de la distance formelle entre les énoncés.

La représentation de la structure d'un gros fichier ***XML*** au moyen du **parser** incorporé est coûteuse en termes de temps de traitement. Le parsage n'est pas monitorisé de sorte que l'on ne peut proposer à l'utilisateur la barre de progression habituelle pointant sur les indices de performance du traitement en cours. L'option qui active le correcteur syntaxique au moment de la frappe, en mode assisté, impose beaucoup de contraites à l'utilisateur. Celui-ci ne pourra pas déplacer librement le curseur dans le fichier édité.

**Remarque** : Nous proposons une fonction expérimentale de récupération des indicateurs ***XML*** pour le cas où, en modifiant la position du point d'insertion, l'utilisateur aura désactivé les commandes d'édition. Cette routine peut traiter au maximum 250 **lignes** éditées, une limite arbitraire qui préserve néanmoins un comportement correct du programme.

La version actuelle du module de calcul des distances formelles entre les énoncés implémente la base de données ***Berkeley*** et le module ***Storable***. Elle lit les indices des énoncés depuis la structure d'arbre implémentée par la bibliothèque graphique ***Tk*** et stocke dans une structure temporaire l'index des formes graphiques extraits du tampon d'édition. Il s'agit du meilleur compromis entre les performances du programme en termes de temps de traitement et les buts réels que nous poursuivons dans le cadre de cette recherche.

Le fait de mémoriser les formes graphiques adressées dans une structure de hachage simple n'est pas nécessairement la meilleure solution informatique. Vu les problèmes de stockage, les structures complexes et volumineuses doivent tenir une place à part dans un fichier ***DB*** dédié. L'accès rapide aux données apporte une amélioration des temps de calcul, tout en conservant la possibilité de traiter des fichiers en édition. La solution que nous proposons représente, en termes de performance et fiabilité, un compromis entre le choix d'un langage de programmation interprété, les performances de l'interface graphique actuelle et les objectifs de notre étude.

## 7.3 Pertinence des indices de proximité

Si l'**indice *Dice*** est calculé à partir des **mots** graphiques considérés comme des unités minimales, la **valeur de tri** retenue comme pertinente varie de **0,7** à **1**, au moment où les valeurs plus grandes seront considérées comme indices valides de la proximité. Pour être rapide et, en l'occurrence, pratique, ce calcul n'est cependant

pas à 100% fiable en raison de la variation morphologique dans les formes lexicales. Calculé à partir des **bigrammes**, ce seuil peut être inscrit dans l'intervalle de **0,5** à **0,7**. Le module procèdera aux superpositions de bigrammes même entre des formes avec un petit score de proximité, ce qui fait que les indices inférieurs à **0,4** ne présentent pas d'intérêt.

La réduction du bruit dans les sorties par le réglage de l'indice de tri a comme conséquence une déperdition des phénomènes de proximité. Notre modèle de calcul des **énoncés proches** est purement fonctionnel. L'aspect du **paradigme des distances** après l'intervention de l'utilisateur sera probablement une conséquence directe des rapports de sens découverts au-delà du décompte des unités minimales.

# 8 Références bibliographiques

CHRISTIANSEN, T., TORKINGTON, N. (1998) : *Perl en action*, Editions O'Reilly, Paris.

DESCARTES, A., BUNCE, T. (2000) : *Perl DBI, le guide du développeur*, Editions O'Reilly, Paris.

GEFFROY, A., LAFON, P., TOURNIER, M. (1974) : *L'indexation minimale, Plaidoyer pour une non-lemmatisation*, Colloque sur l'analyse des corpus linguistiques : « Problèmes et méthodes de l'indexation minimale », Strasbourg 21-23 mai 1973.

MENARD, N. (1983) : *Mesure de la richesse lexicale, théorie et vérifications expérimentales*, Slatkine-Champion, Paris.

MEYER, B. (2000) : *Conception et programmation orientées objet*, Editions Eyrolles, Paris.

MOSUT, C.I. (2003) : *L'évolution des représentations médiatiques du premier gouvernement roumain post-communiste. Approche énonciative et lexicométrique comparée des modes d'argumentatio dans un corpus d'articles de presse française (1989-1990)*, thèse de doctorat, Université Paris III.

REINERT, M. (1990) : *Alceste, Une méthodologie d'analyse des données textuelles et une Application : Aurélia de Gérard de Nerval* in *Bulletin de Méthodologie Sociologique*, n°26.

SALEM, A. (1993) : *Méthodes de la statistique textuelle*, Thèse d'Etat, Université Sorbonne Nouvelle (Paris 3).

WALL L., CHRISTIANSEN T., SCHWARTZ, L. R. (1996) : *Programmation en Perl*, Editions O'Reilly, Paris.

WALSH N. (2000) : *Introduction à Perl/Tk. Interfaces graphiques avec Perl*, Editions O'Reilly, Paris.