



MkCorpus/CorpusPlusBuilder v1.04

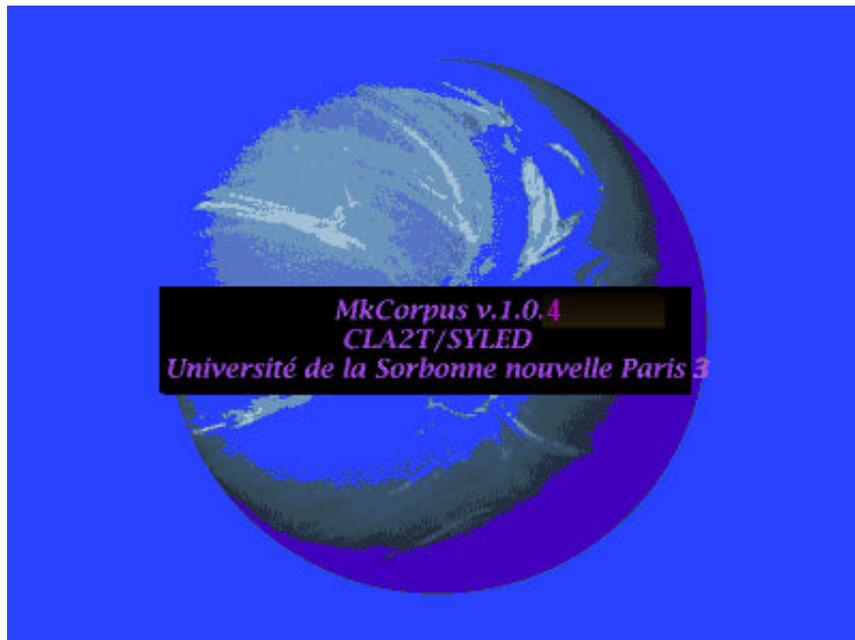
Outil de préparation et de manipulation de Corpus

CLA2T/SYLED

Page Web MKCORPUS (téléchargement, documentation) :

www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/mkcorpusProject.htm

Document de travail (Juillet 2003)



ILPGA/Sorbonne nouvelle - Paris 3
19 rue des Bernardins, 75005 Paris
Tél : 01.44.32.05.75

Email : fleury@msh-paris.fr
Ou serge.fleury@univ-paris3.fr

Hypertoile SF :
<http://www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/>

Hypertoile TAL Paris 3 :
<http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/>

Sommaire

1	Préambule	3
2	Installation	4
3	Mises à jour	5
4	Modules "originaux" de MKCORPUS.....	6
4.1	Mise en œuvre et utilisation des modules	6
4.1.1	Les menus.....	6
4.1.1.1	Menu FICHER	6
4.1.1.2	Menu EDITION.....	7
4.1.1.3	Menu SEARCH	7
4.1.1.4	Menu MARKUP <[^>]+>	7
4.1.1.5	Menu HTML	7
4.1.1.6	Menu XML.....	7
4.1.1.7	Menu SGML.....	8
4.1.1.8	Menu MAP_CORPUS	8
4.1.1.9	Menu CORPUS	11
4.1.1.10	Menu NLP	13
4.1.1.11	Menu TYPWEB.....	13
4.1.2	Les boutons	15
5	Références bibliographiques	16
6	Annexes	17
6.1	Aide Webxref.....	17
6.2	Aide Typweb.....	21

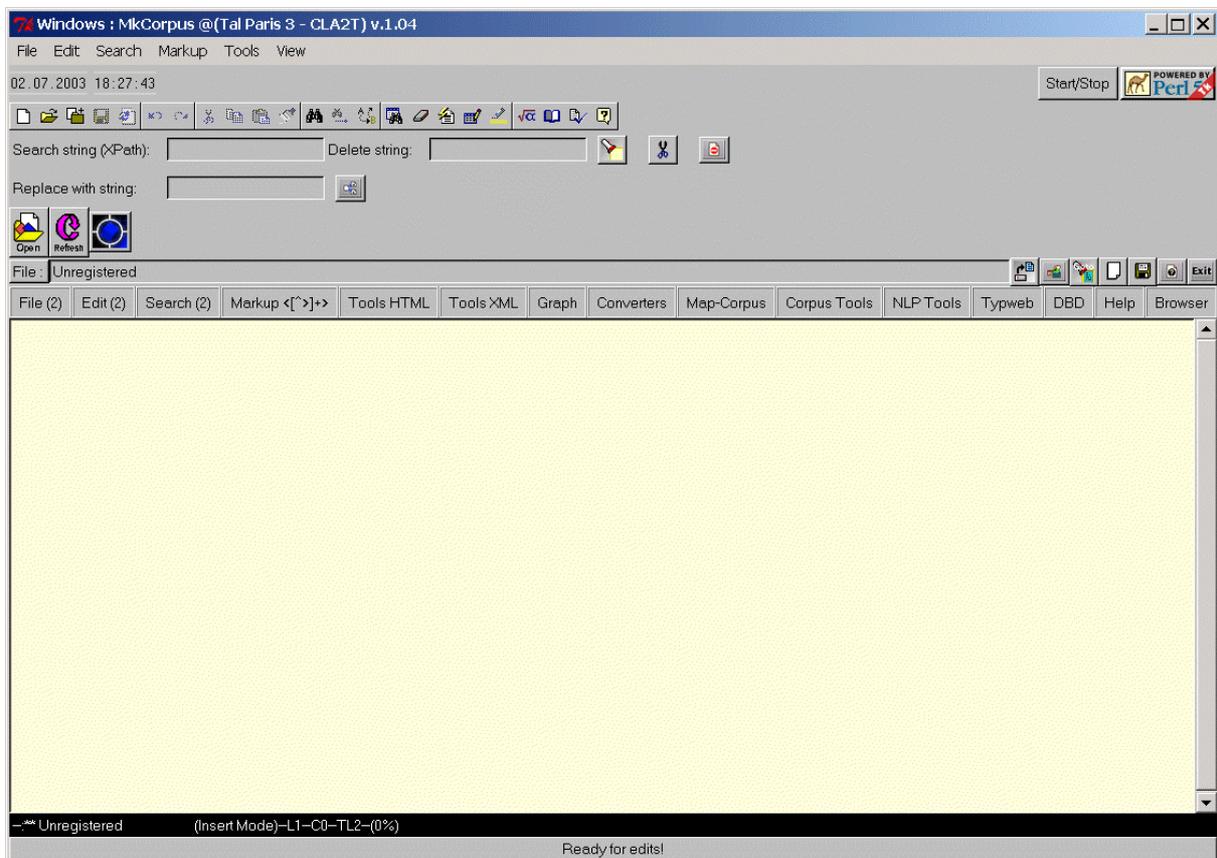
1 Préambule

Mkcorpus est un programme de préparation de corpus pour leurs analyses ultérieures via des outils traditionnels du TAL. Il est écrit en Perl/TK.

Ce programme permet :

- ?? de visualiser le corpus,
- ?? de manipuler via des outils idoines le contenu du corpus et de ses éléments pour les formater suivant les contingences imposées par les outils (suppression de balises, nettoyage...).

Cet outil se présente comme un éditeur traditionnel et les menus construits permettent de réaliser des opérations sur les fichiers visualisés dans la zone d'édition ou attachés aux programmes de traitement.



2 Installation

Pour utiliser MKCORPUS, il faut télécharger :

- ?? soit le **programme d'installation** disponible pour une version de MkCorpus pour Windows
- ?? soit **l'archive source** (le code complet de l'application)

Le tout est disponible ici :

www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/mkcorpusProject.htm

⚡ La version pour Windows s'installe en exécutant le *setup* contenu dans l'archive. Après installation, **il faut installer le correctif** : (1.04) <http://www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/mkcorpus/Correctif-win32-1.04.zip>, (versions antérieures) <http://www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/mkcorpus/Correctif-win32-1.zip>, dans cette archive, vous trouverez un readme qui vous indique où installer 2 fichiers supplémentaires.

Pré-requis : A priori aucun... à part Windows. Cependant, certains modules (non recensés ici) ont besoin de Perl pour leur exécution. Si vous disposez de Perl sur votre propre machine, MkCorpus sera complètement opérationnel. Si vous souhaitez installer Perl, il suffit de le récupérer sur le site de activestate.com par exemple.

⚡ L'archive source contient toutes les ressources disponibles (le code source complet). Le plus simple pour l'installer est de "dézipper" l'archive dans un répertoire en respectant l'arborescence construite dans l'archive. Cette version a été testée sous Windows et Linux. Pour lancer MKCORPUS, il faut lancer le programme Perl nommé `loadMkCorpus.pl` ou le programme `win32MkCorpus.pl` sous Windows.

Pré-requis : il faut disposer de Perl et de Perl/Tk. Il faut aussi installer les modules Perl contenus dans le répertoire `mkCorpusModules` contenu dans l'archive. Pour ces installations suivre les instructions contenues dans les `Readme` de chaque module : il faut en général déposer une bibliothèque dans le répertoire local `lib` de Perl (en général `c:\Perl\lib` sous windows, sous linux, la procédure d'installation standard fait le travail parfaitement).

3 Mises à jour

La documentation des modules originaux de MkCorpus est donnée infra. Les modules complémentaires qui ont conduit à des mises à jour seront présentés dans cette partie. On trouvera aussi en ligne un descriptif de ces mises à jour.

- ?? **Mars-Juin 2003** : ajout de fonctionnalités XML, module Graphique via Graphviz (Menu Graph), générateur de corpus XML : txt2xml (menu XML).
- ?? **Janvier-février 2003** : quelques correctifs liés aux modules XML.
- ?? **Octobre/novembre 2002** : intégration de plusieurs fonctionnalités : (1) mise en place du KUB (v.1, documentation à venir) (2) processeur XSLT (documentation à venir).
 - ?? Installation de XML::LibXML, XML::LibXSLT
 - ?? On se reportera à la présentation faite infra pour l'installation de Sablotron : procédure identique à partir des mêmes "repositories"
- ?? **Été 2002** : intégration de plusieurs fonctionnalités : [Directory2XML](#), [Collocation in XMLDoc via XPath](#), [Extraction aléatoire de fichiers dans une arborescence](#).
- ?? **Mai 2002** : intégration des bibliothèques TextTools et MarkupTools développées par Calin Mosut pour edXML : [EDXML](#), [TextTools](#), [MarkupTools](#)
- ?? **3 février 2002** : Ajout d'un module pour traiter les fichiers WORD (v.1.02) : Word2Lexico3 ([HELP module](#)) et Word2Xml ([HELP module](#))
- ?? **30 Janvier 2002** : Ajout d'un module XSLT (v.1.02) : XML -> HTML via XML::Sablotron ([HELP module](#))
 - ?? Installation de XML::Sablotron
 - ?? C:\> ppm
 - ?? ppm\>set repository RK http://theoryx5.uwinnipeg.ca/ppmpackages/
 - ?? ppm\> set save
 - ?? ppm\> install XML::Sablotron
- ?? **05 Janvier 2002** : Modification du module XML->XPath ([HELP module](#))
- ?? **03 Janvier 2002** : ajout d'un module Cordial2xml : transcodage d'un état de sortie étiqueté de Cordial (cf paramétrage infra) vers XML ([HELP module](#))
- ?? **02 Janvier 2002** : ajout d'un browser (via tkWeb) : lecture documentation en ligne, sortie HTML à venir...
- ?? **Décembre 2001** : ajout d'un module de concaténation de fichiers ([HELP module](#))
- ?? Pour le reste voir infra

4 Modules "originaux" de MKCORPUS

On donne ci-dessous un descriptif des différentes opérations actuellement disponibles via cet outil. Ces informations seront complétées par les descriptifs supplémentaires disponibles sur la page web de MkCorpus.

Les modules présentés ci-dessous sont disponibles actuellement :

- ≡≡ **Fichier** : ouverture, sauvegarde
- ≡≡ **Edition** : outils traditionnels d'édition
- ≡≡ **Search** : recherche (regexp), remplacement, (dans menu et via boutons)
- ≡≡ **Markups** <[^>]+>: liste, suppression, modification
 - Ce module utilise la représentation de balise sous la forme d'expression régulière du type <[^>]+> : solution intéressante mais pas toujours concluante.
 - Extraction de texte entre balises <[^>]+>.
- ≡≡ **Tools HTML** :
 - En développement...Des modules de transcodage sont disponibles.
- ≡≡ **Tools SGML** :
 - En développement...
- ≡≡ **Tools XML** :
 - Outils spécifiques pour travailler sur des documents XML: passage, représentation graphique, stat, commentaires, extraction de texte entre balises, processeur XSLT...
- ≡≡ **Map Corpus** :
 - Manipulation du documents XML représentés sous la forme de cartes graphiques ou d'arbres
 - Représentation graphique de corpus balisés : "parcours" dans l'arbre associé au texte balisé via une interface graphique
- ≡≡ **Corpus Tools** :
 - Préparation de corpus (recherche et nettoyage de caractères), avec sous-modules spécifiques pour Lexico, Cordial, Alceste
 - Statistiques sur fichiers, concaténation de fichiers
 - Conversion de fichiers Word vers Lexico3 ou Xml
- ≡≡ **Typweb** : la chaîne typweb est disponible complètement : version 036 et 038, avec les programmes associés. La phase d'aspiration de site est en cours de mise au point.
- ≡≡ **NLP tools** : concordance, bigrammes, collocation, programme `xword` (cf "Web programming with perl" O'Reilly)

4.1 Mise en œuvre et utilisation des modules

MKCORPUS se présente comme un éditeur de texte traditionnel : une fenêtre d'édition, des menus et des boutons. La fenêtre principale est donc destinée à abriter un fichier en édition. Les principales actions disponibles sont activées via les menus ou via les boutons (qui correspondent en fait à des raccourcis de fonctions disponibles dans les menus).

4.1.1 Les menus

4.1.1.1 Menu FICHIER

- ≡≡ *View dir* : affiche le contenu de l'arborescence du disque de travail sous une forme d'arbre.
- ≡≡ *Select File & open* : déclenche l'ouverture d'une boîte de dialogue pour l'ouverture d'un fichier afin de l'éditer.
- ≡≡ *Select File & not open* : déclenche l'ouverture d'une boîte de dialogue pour associer un fichier à des traitements sans ouvrir celui ci dans la fenêtre principale. Cette option est surtout destinée à être utilisée dans le cas de travail sur de gros fichiers. Cette option n'est pas encore disponible.
- ≡≡ *Open this file* : charge le fichier courant, celui dont le nom est affiché dans la boîte File.

- ⚡⚡ *Save this file* : sauvegarde le fichier courant, celui dont le nom est affiché dans la boîte File.
- ⚡⚡ *Save this file as* : sauvegarde le fichier courant, celui dont le nom est affiché dans la boîte File avec possibilité de le renommer. Cette option fait appel à une boîte de dialogue.
- ⚡⚡ *Clear* : cette option crée un nouveau fichier dans la fenêtre principale. Ce fichier n'existe pas physiquement. Il faut au préalable l'enregistrer.

4.1.1.2 *Menu EDITION*

Ce menu fait appel à des opérations traditionnelles disponibles dans les éditeurs : copier, coller, sélectionner...

4.1.1.3 *Menu SEARCH*

Ce menu fait appel à des opérations traditionnelles disponibles dans les éditeurs : recherche incrémentale, recherche globale, remplacement... Ces recherches utilisent la notions d'expression régulière. Le motif de recherche doit être au préalable intégré dans le champ "Search String" idem pour le motif de remplacement le cas échéant.

- ⚡⚡ Le bouton "highlight" est équivalent à une recherche globale du motif donné.
- ⚡⚡ Le bouton "delete" est équivalent à un remplacement global du motif donné par le motif de remplacement donné.
- ⚡⚡ Chaque recherche concluant provoque l'affichage coloré et clignotant du motif trouvé. Le bouton "no highlight" inhibe ce formatage.

4.1.1.4 *Menu MARKUP <[^>]+>*

Ce menu permet de réaliser des traitements sur des fichiers balisés, avec des balises du type <BALISE>. Les opérations disponibles utilisent une représentation formelle de ce type de balise sous la forme d'une expression régulière suivante "<[^>]+>".

Remarque importante : la représentation d'une balise sous la forme de cette expression régulière n'est en aucun cas une solution optimale pour représenter une balise. Nous l'avons retenu dans la mesure où les fichiers que nous avons l'habitude de manipuler ne présentent pas de balises dont l'écriture est scindée sur deux lignes.

- ⚡⚡ *View all* : crée une nouvelle fenêtre contenant toutes les balises du fichier courant de la fenêtre principale. Il est possible de sauvegarder le fichier construit.
- ⚡⚡ *delete all* : supprime toutes les balises du fichier courant de la fenêtre principale.
- ⚡⚡ *global search* : recherche globale de toutes les balises.
- ⚡⚡ *I-search* : recherche incrémentale de motifs donnés dans le champ de recherche.
- ⚡⚡ *repeat I-Search* : nouvelle recherche du motif en cours de recherche.
- ⚡⚡ *Extract text in markup* : cette option active la génération d'une nouvelle fenêtre qui permet de sélectionner des balises du texte courant pour : les modifier, les supprimer ou bien pour extraire le contenu situé entre la balise ouvrante et la balise fermante de la balise visée.

4.1.1.5 *Menu HTML*

- ⚡⚡ *HTML-Convertir* : ce sous-menu contient des programmes de transcodage.
- ⚡⚡ *HTML-Tidy* : ce sous-menu permet l'utilisation de l'utilitaire Tidy¹ sur des pages HTML locales. Cette option prend en entrée un répertoire contenant un site et réalise le nettoyage des fichiers HTML suivant la configuration de Tidy contenu dans le fichier Tidy/config-html.txt

4.1.1.6 *Menu XML*

¹ Développé par Dave Raggett, HTML Tidy est proposé par le W3C. Sa finalité première est de nettoyer les fichiers HTML des erreurs (ex. éléments qui se chevauchent) et de tout ce qui est interdit par les recommandations HTML (<http://www.w3c.org/People/Raggett/tidy/>).

(Outils XML, documentation complète à venir)

- ⚡ XML-Viewer : outils pour visualiser l'arbre XML associé au document XML contenu dans la fenêtre principale
- ⚡ XML-Parser : outil pour parser le document XML contenu dans la fenêtre principale
- ⚡ XML Xpath : rechercher et extraire des parties du document XML via Xpath : en indiquant le chemin Xpath souhaité dans la zone de recherche, le programme produit, s'il le trouve, le contenu de la zone trouvée dans une zone d'édition qu'il est possible de sauvegarder.
- ⚡ XML-Grove : en développement
- ⚡ XML-ParseDTD : en développement
- ⚡ XML-Extractor : outils pour réaliser des extractions des contenus d'éléments sélectionnés (plusieurs options sont disponibles)
- ⚡ XML-Element : plusieurs outils sont disponibles ici : afficher tous les commentaires du document XML contenu dans la fenêtre principale, afficher des statistiques sur le document XML contenu dans la fenêtre principale
- ⚡ XML-AddTagInTextNode : programme permettant d'ajouter des nœuds au document XML chargé
- ⚡ XML-TIDY : conversion en format XHTML de sites web initialement codés en HTML
- ⚡ XML-Converters : programmes de transcodage

4.1.1.7 Menu SGML

- ⚡ Module en cours de test et non documenté.

4.1.1.8 Menu MAP_CORPUS

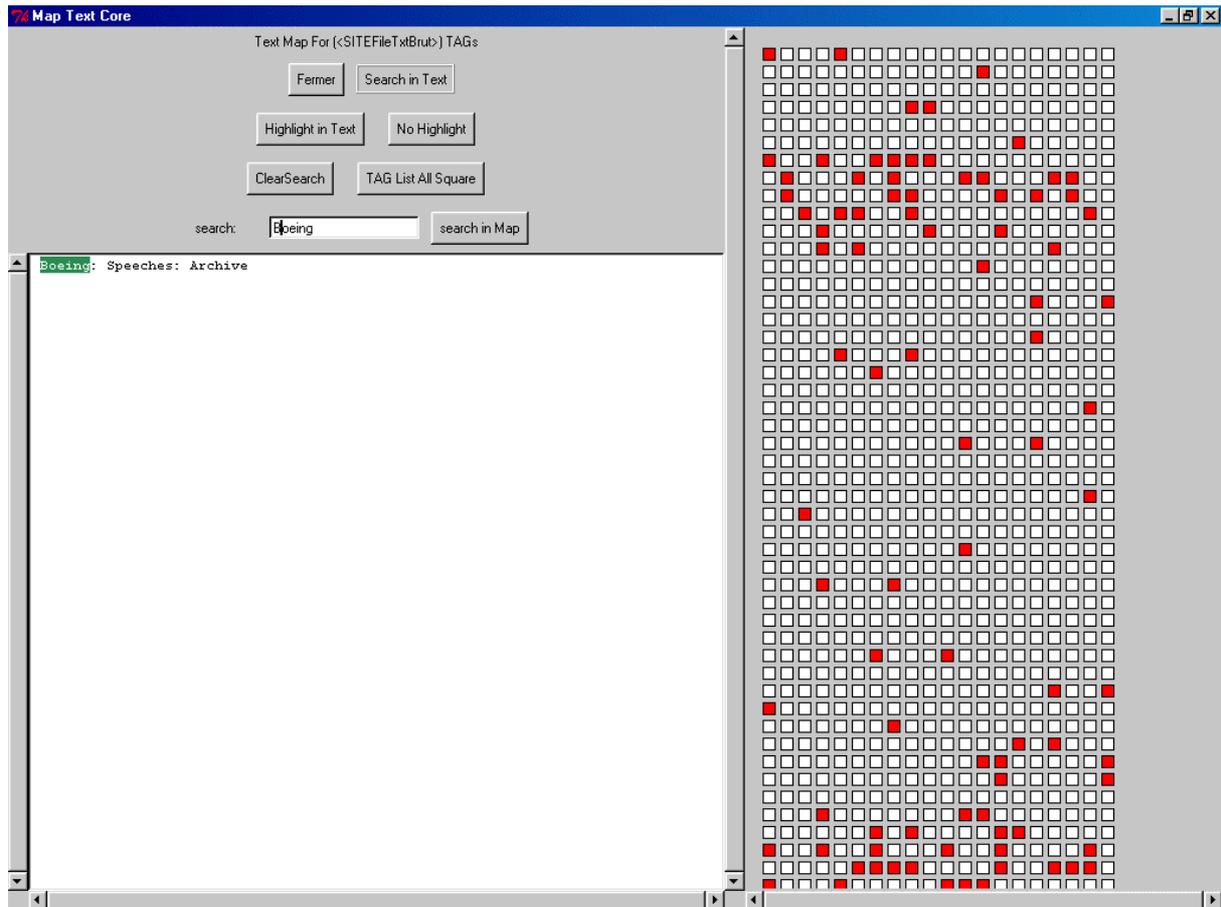
- ⚡ Make TAG-MAP (REGEXP)

Cette option s'inspire de la notion de "carte graphique de textes" disponibles dans Lexico3. L'enjeu est le suivant : il s'agit de donner une représentation graphique d'un document XML, sous la forme d'une carte de carrés colorés, sur la base d'une sélection d'un certain niveau de représentation (sélectionné par l'utilisateur) de ce document XML. On peut aussi considérer que cette option permet de se "promener" dans l'arbre XML du document visé en sélectionnant le niveau des nœuds de l'arbre à visualiser, on peut ensuite poursuivre la "descente de l'arbre" en sélectionnant un niveau de nœud plus profond. On donne une illustration de cette option dans les figures qui suivent.

La représentation graphique produite du document est réalisée en utilisant des expressions régulières.

Dans cette figure, on suppose que MKCORPUS a chargé un corpus XML contenant des zones textuelles comprises entre les balises <SITEFILEXTBRUT> et </SITEFILEXTBRUT>. L'activation de l'option MapXMLCorpus déclenche la génération dans un menu déroulant de toutes les balises du corpus. On peut ensuite sélectionner la balise que l'on souhaite visualiser sous la forme d'une carte graphique. Dans notre exemple, la balise <SITEFILEXTBRUT> a été sélectionnée. MKCORPUS se charge ensuite de construire une carte de ces zones. Dans la figure, chaque carré construit correspond à une zone textuelle associée à la balise choisie (balises ouvrante et fermante). On peut ensuite réaliser différentes opérations :

- (1) Rechercher des éléments textuels dans la carte ; pour cela, il convient de donner une chaîne de caractère dans la zone de recherche puis d'activer le bouton "search in map", les zones textuelles rouges contiennent la chaîne visée, les autres, en blanc, ne le contiennent pas. Dans notre figure, le mot Boeing a été mis en valeur dans la carte, les carrés rouges contiennent donc ce mot.
- (2) Afficher le contenu textuel d'un carré de la carte ; en cliquant sur le bouton gauche de la souris au dessus du carré, le contenu textuel apparaît dans la zone d'édition.
- (3) Mettre en valeur une chaîne de caractères dans la zone d'édition : mécanisme Highlight déjà vu plus haut. Dans notre figure, le mot Boeing a été mis en valeur dans la zone d'édition.
- (4) Sélectionner une balise fils de l'un des carrés de la carte et la sélectionner pour générer une nouvelle carte : le clic droit sur un carré de la carte déclenche la génération de toutes les balises présentés dans cette zone textuelle. La sélection de balises peut ensuite déclencher la génération d'une nouvelle carte sur la base de cette nouvelle sélection.



☞☞ Make TAG-MAP (DOM)

Cette option est une extension de la précédente qui utilise DOM pour construire la représentation graphique du document sous la forme de carrés associés à un ou plusieurs éléments de l'arbre.

☞☞ Make TREE-MAP (DOM)

Cette option prend elle aussi appui sur DOM pour construire, non plus une carte d'un certain type d'éléments du document XML (options précédentes), mais l'intégralité du document représenté sous la forme d'un arbre. La figure qui suit donne un exemple d'arbre construit à partir d'un document XML chargé au préalable dans MKCORPUS.

3. Il existe aussi une option qui permet d'appliquer des modifications sur tous les caractères du fichier sur la base d'une table de transcodage contenue dans le fichier nommée TableCharacter.txt. Il est possible de modifier cette table, chaque ligne est construite sous la forme suivante :

`<caractèreInput><caractèreOutput><chiffre>`

Le premier caractère sera remplacé par le second quand cette option est activée. Tous les caractères non contenus dans cette table seront remplacés par un blanc.

☞☞ PrepHtml2Txt

Ces items permet de transformer des pages HTML en fichiers texte.

Deux options sont disponibles :

1. la première permet de réaliser le transcodage sur un répertoire complet (de manière récursive). En sortie, on obtient un fichier réécrit pour chaque fichier lu dans l'arborescence parcourue et un fichier global concaténant l'ensemble des fichiers transcodés.
2. le seconde réalise l'opération de transcodage sur le fichier lu dans la fenêtre d'édition active.

Ces programmes ont été inspirés par ceux disponibles à l'adresse suivante :

<http://www.codearchive.com/home/jon/>.

☞☞ Option Lexico

Ce sous-menu permet de construire des corpus prêts à être analysés par Lexico3 après sélection d'élément (plusieurs options sont disponibles).

Il permet aussi de lancer des traitements particuliers : remplacement de tous les caractères majuscule par le même caractère en minuscule précédé par une étoile (cf documentation Lexico3), reformatage des sorties produites par Lexico2.

☞☞ Option Cordial :

Une opération disponible actuellement (Make-Corpus Tag (For Lexico v1, v2)) concerne le traitements des résultats issus de Cordial Analyseur. Il est possible de formater les résultats issus de Cordial et de les préparer pour être traitées par Lexico.

Le fichier à soumettre à Cordial doit être balisé sous la forme suivante :

```

<balise1=valeur1>
zone textuelle 1
<balise2=valeur2>
zone textuelle 2
...
```

S'il y a des balises dans le texte, il faut qu'elles aient l'allure précédente, s'il n'y en a pas, aucun problème a priori. Le résultat créé par Cordial est ensuite reformaté pour Lexico.

Le résultat produit par l'étiquetage est de nouveau soumis à MKCORPUS qui se charge de remettre en forme ce résultat d'étiquetage pour qu'il soit de nouveau utilisable dans LEXICO. Ce travail de reformatage produit en fait 6 fichiers qui peuvent être tous traités par LEXICO :

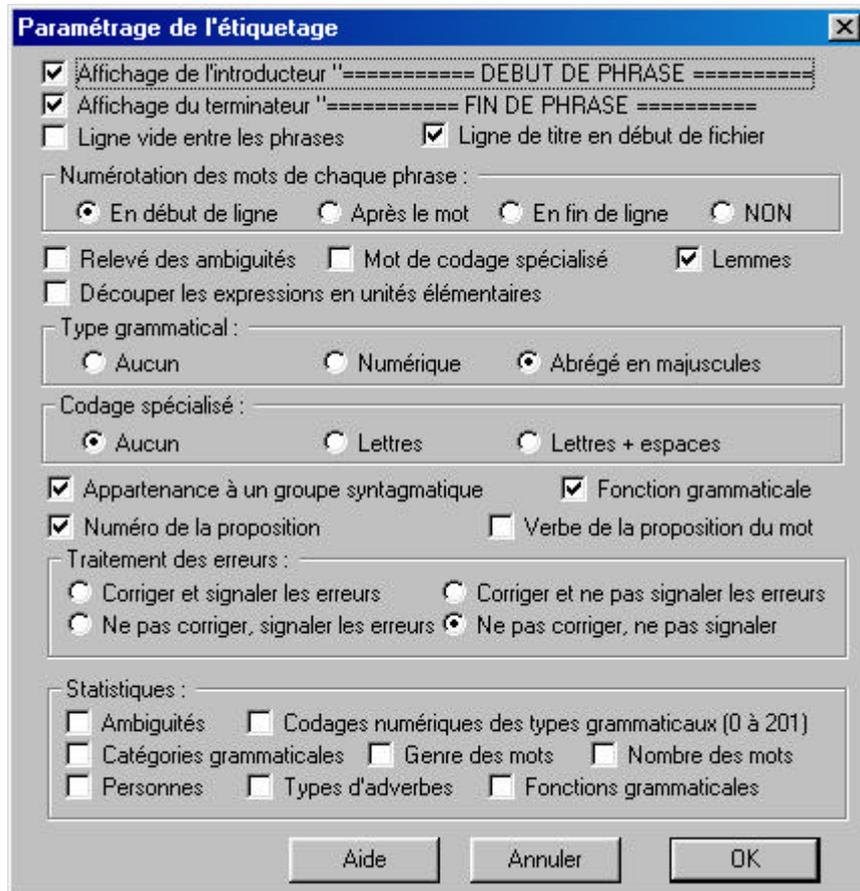
- ?? Un fichier contenant toutes les formes graphiques (i.e. le corpus initial)
- ?? Un fichier contenant tous les lemmes
- ?? Un fichier contenant toutes les étiquettes syntaxiques

?? Un fichier contenant les couples (lemme, étiquette)

?? Un fichier contenant les couples (forme, étiquette)

?? Enfin, un fichier contenant la concaténation des trois premiers fichiers : i.e. une partition d'un même texte sous trois facettes complémentaires. On présentes un extrait de cet état du corpus ci-dessous :

Le paramétrage de cordial à utiliser est le suivant :



Les options "Cordial" de ce menu permettent ensuite de remettre en forme les résultats de Cordial. On obtient en fait plusieurs sorties : formes, lemmes, catégories, forme_catégorie, lemme_catégorie et une version du corpus qui contient une partition regroupant les lemmes, une partition regroupant les formes et une partition regroupant les catégories.

4.1.1.10 Menu NLP

Ce menu regroupe différents programmes de manipulation pour les textes manipulés (documentation complète à venir) :

- ☞☞ Concordance
- ☞☞ Collocation
- ☞☞ Bigramme
- ☞☞ Fourgramme
- ☞☞ Wordcount
- ☞☞ xword

4.1.1.11 Menu TYPWEB

Les outils Typweb sont intégrés dans ce menu. Pour un descriptif de ces outils : <http://www.cavi.univ-paris3.fr/ilpga/ilpga/sfleury/typweb.htm>

Chaîne Typweb version 036

- *webxref*

On peut activer webxref sur un site aspiré localement et contenu dans un répertoire donné.

On peut activer webxref sur un fichier index.htm sauvegardé localement et contenu dans un répertoire donné. Il faut au préalable charger le fichier dans la fenêtre principale.

- *mktipo*

On peut activer le programme mktipo présenté supra dans ce menu.

- *mkstat*

On peut activer le programme ExtAndStatFrCorpTwb présenté supra dans ce menu.

Chaîne Typweb version 038

- *webxref*

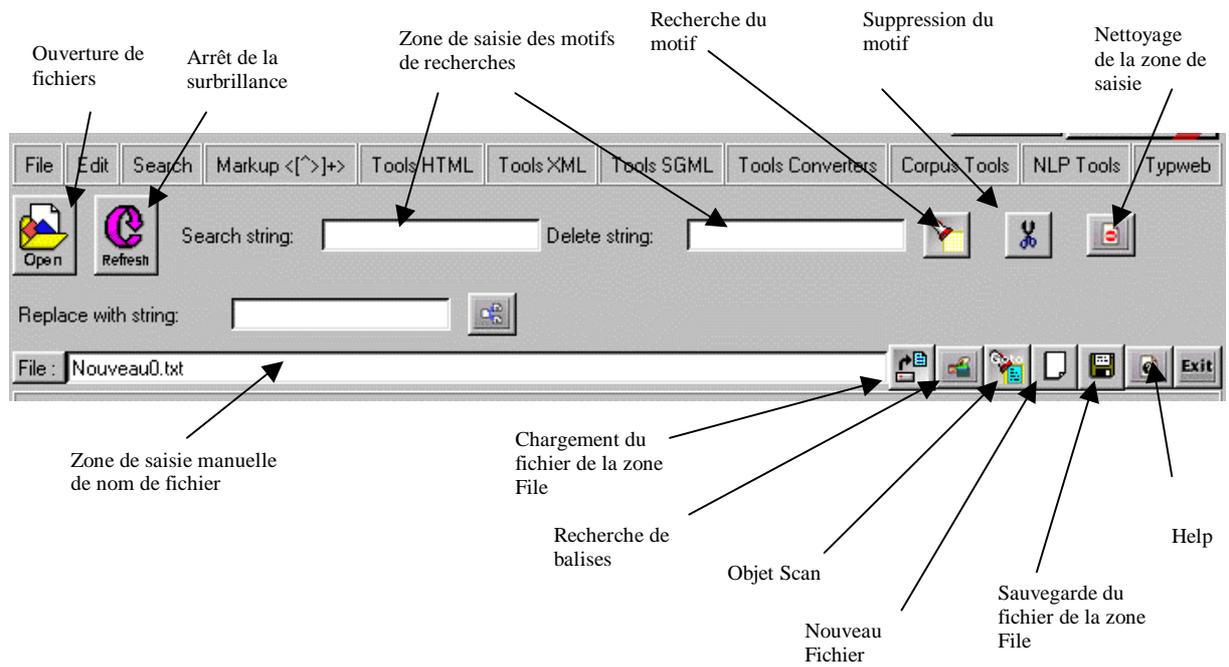
On peut activer webxref sur un site aspiré localement et contenu dans un répertoire donné.

On peut activer webxref sur un fichier index.htm sauvegardé localement et contenu dans un répertoire donné. Il faut au préalable charger le fichier dans la fenêtre principale. Certaines options ne sont disponibles que sous Unix.

On peut, à partir des résultats fournis par les différentes options de webxref, générer les fichiers de travail de la chaîne Typweb : matrice de liens, corpus de TAGs, matrice de mots et de TAGs.

Toutes les opérations disponibles sont décrites dans les fichiers d'aide fournis dans cette version.

4.1.2 Les boutons



5 Références bibliographiques

- ⌘ **Brown Martin**, Perl Annotated Archives, Ed. Osbourne/Mc Grawhill
- ⌘ **Brown Martin C.**, XML Processing with Perl, Python, and PHP , SYBEX, 2002.
- ⌘ **Christiansen Tom & Nathan Torkington**, Perl cookbook, O'Reilly
- ⌘ **Clinton Wrong**, Web Programming with Perl, O'Reilly
- ⌘ **Glover Mike, Humphreys, Ed Weiss**, Perl 5 how-to, The Wait Group, Inc.
- ⌘ **Orwant John**, (édité par), "Computer Science & Perl Programming : Best of The Perl Journal", 600 pages, édition O'Reilly, ISBN 0-596-00310-2, 2002
- ⌘ **Orwant John**, (édité par), "Web, Graphics & Perl/Tk : Best of The Perl Journal", 504 pages, édition O'Reilly, ISBN 0-596-00311-0, 2002
- ⌘ **Pierrestéguy Frédéric**, Bien Débuter avec GNU Emacs, Masson
- ⌘ **Ray Erick T., Jason McIntosh**, Perl & XML, XML processing with Perl, édition O'Reilly, 2002.
- ⌘ **Ray Erick T., Christopher R. Maden**, Learning XML, édition O'Reilly, ISBN 0-596-00046-4, 2001.
- ⌘ **Riehl Mark, Sterin Ilya**, XML and Perl, New Riders Publishing 2003
- ⌘ **Wall L. & al.** , Programmation en Perl, Traduction française, 2^{ème} édition, O'Reilly
- ⌘ **Walsh Nancy**, Introduction à Perl/Tk, O'Reilly

6 Annexes

6.1 Aide Webxref

S. Fleury
01/07/2001

Les options disponibles dans le menu Typweb :

WEBXREF 036

1. Read One Local Web Dir (036) : cette option permet d'exécuter webxref sur un répertoire contenant un site web localement aspiré
2. Read Index File (036) : cette option permet d'exécuter webxref sur un répertoire contenant un site web localement aspiré, le site visé est celui qui est associé au fichier préalablement chargé dans la fenêtre d'édition de mkCorpus

WEBXREF 038

1. Read One Local Web Dir (038) : cette option permet d'exécuter webxref sur un répertoire contenant un site web localement aspiré
2. Read One Dir of Local Web Dir (038) : cette option permet d'exécuter webxref sur un répertoire contenant plusieurs sites web localement aspirés
3. Read Index File (038) : cette option permet d'exécuter webxref sur un répertoire contenant un site web localement aspiré, le site visé est celui qui est associé au fichier préalablement chargé dans la fenêtre d'édition de mkCorpus
4. Read One Local Web Dir (038 Unix Lynx) : idem (1) avec sorties Lynx
5. Read One Dir of Local Web Dir (038 Unix Lynx) : idem (2) avec sortie Lynx
6. Read Index File (038 Unix Lynx) : idem (3) avec sortie Lynx
7. Read Local Web Dir (038-homolWin32) : idem (1)
8. Read Index File (038-homolWin32) : idem (2)
9. Read Local Web Dir (038-homolUnix Lynx) : idem (1) avec sortie Lynx
10. Read Index File (038-homolUnix Lynx) : idem (2) avec sortie Lynx

WEBXREF DOC 038 :

Usage: webxref -options file.html || webxref -options site/
 then navigate to "working_directory/res[](site)/"

 webxref -options sitel site2
 then see "res[](sitel)" and "res[](site2)"

 webxref -at "path" file.html
 then navigate to "path/res[](site)/"

Options: -help/-h -noxref -xref/-x -onexref -fluff -htmlonly
 -rep -norep -at -http -delay seconds
 -silent/-s -verbose/-v -errors/-e -noint
 -spell -html -del -lynx -brief -fullpath
 -islocal <address> -avoid/-a <regexp>
 -one/-1 -depth <depth> -rappspec <number>
 -date <yymmdd> -time <hhmmss> -before -after
 -find <string> -findexpr <regexp>
 -replace <string> -replaceexpr <regexp> -by <string/expr>

=====
Which parameters to use for what purpose:
=====

Webxref checks the given file and follows the links in that file. While working it lets you know it's alive by printing to STDOUT verbose messages. It also prints in the report file a '+' for each file checked ok, and a '-' for each file with a problem.

Default webxref gives for each file found on your local disks a report on its headers,tag elements(with attributes-values) and links. After parsing

it as a string, the routine DissectFile will display data from the HTML syntax tree. The routine was inspired by the htmlscript "dissectsite.hts" found at "http://worldwidemart.com/scripts/htmlscript/dissect/". Specify `-norep` to discharge it or see the main section for configuring.

A webxref run can take some time. You can, however, interrupt webxref with `ctrl-c` (Unix). Webxref will report only the files it has inspected up to that moment and exit. (*New!*)(Note: this is not reliable! webxref is not interruptable at any time, due to the C-libraries not being re-entrant. (This probably does not interest you at all, but it's not the author's fault.)) Specify `-noint` if you don't want webxref to try and generate output after an interrupt.

When the whole site has been searched, all links have been inspected and all its .html, .htm files found have been dissected, webxref prints a report. Actual default is a long report in .txt form. The option `-html`, lets you change the form default. The option `-at`, allows you to choose a directory on your disk to put the results in. See also the examples.

If you want more information while webxref is working specify `-verbose` to get messages on every file or `-errors` to see only files with problems. With `-silent` webxref prints few messages while working.

Webxref keeps track of which html-documents are being linked to from other documents. This is called cross-referencing, hence webxref's name. If you are not interested in this, specify `-noxref`, so you won't be told where things have failed and probably have to run webxref again. If you're just interested in one location where a file is referenced specify `-onexref`. This saves memory too.

If you need to know if there are files and/or directories in your site that are not referenced at all by any pages in your site specify `-fluff`.

If you want to only inspect files that really have the .html or .htm extension specify `-htmlonly`

If you specify `-fullpath`, you'll get the full paths for files. Default, the file names will be abbreviated: /u/people/rick/www/a.html is printed as "a.html" (webxref is called from ~/rick/www).

If you use full URLs in your site referring to your own site, say "www.sara.nl" is your www-address and you use links like `` then tell webxref that "www.sara.nl" actually can be found on the local machine with: `-islocal 'www.sara.nl'`

If you want to avoid certain files use the `-avoid` parameter to specify which files to avoid.

If you want to limit the number of files webxref inspects you may want to limit the scan to 1 or 2 directories deep in the file system. If you specify `-depth 0` only files in the current directory are inspected.

If you just want to check if links in a file are valid specify `-one` (or `-1`). Only the links present in the file are tested, but no more. Use this with `-files` to specify a collection of files to just check those files.

Specify `-http` if you want webxref to check if the http:// links work. After all local files are inspected. This may be time-consuming. To avoid overloading a webserver there is a delay of 1 second between checks. If you want longer or shorter delays specify the number of seconds with `-delay`. (Longer delays may be necessary if a lot of links refer to the same webserver.)

To see if you have files or directories that were modified last before or after a certain date/time use: `-before/-after -date yymmdd -time hhmmss`. If `-before` is given files are reported that were modified before the date given, with `-after` files last modified after the date given are reported.

Default, simply list the file or directories at the end of the command. To tell webxref which files to inspect use `-files` or `-f`. Webxref generates different results directories only if the files given as arguments are from different sites.

Webxref can search and even search-replace text, see later.

=====
What the parameters do:
=====

While checking webxref prints messages to STDOUT according to:

`-silent/-s` Few messages, list problems at the end of the run.

```
-verbose/-v    Print information while checking files.
-errors/-e     Print errors when they occur, even when -silent.
```

Webxref generates a report according to:

```
-noint        Do not generate output on interrupt
-norep        The routine elements is discharged
-spell        Checks html files for syntax errors
-brief        List just problems.
-xref/-x      List which files reference files (cross-references).
-noxref       Do not list which files reference files (default).
-htm1         Print report in .html form.
-del          Delete Html Report Files
-lynx         Generates Lynx Dump on each html file (for the XML corpus)
-rappspec nb The nb is used in the name of the report directory
-at           Lets you choose a directory to put the results in
```

Webxref inspects files/directories according to:

```
-fluff        List which files/directories are never used.
-htm1only     Only inspect files with the .html/.htm extension.

-fullpath     Print full-length filenames.
-islocal url  'www.mymachine.nl' is actually a local file reference.
-avoid regexp Avoid files with names matching regexp for inspection.
-depth number The maximum directory nesting level.
              0 means: current directory only,
              1 means: directories from the current directory.
              100 probably means there is no restriction in
              how deep webxref is allowed to find files.
-one/-1       Specify -one if you just want to check the links
              from the given file(s) and no further link following.
-htp         Check external URLs via the network.
-delay seconds Wait the specified number of seconds between HTTP checks
-date -time   Date [yymm<dd>], time [hhmm<ss>].
-before -after List files that are modified before or after
              the date/time given with -date and -time.
```

```
=====
Find/replacement: ** EXPERT ONLY **
=====
```

Webxref can scan your site for files containing certain text. To find fixed text use `-find`. To find text using e.g. wildcards use `-findexpr`. The Perl expression is matched with the text of the file under test. Take care to not have the shell interpret '*' and '/' by using appropriate quoting. Search is always case-insensitive. Webxref does search/replace beyond end-of-line. I.e. newlines are matched, and can even be inserted (use `\n`).

To replace text with something else use `-replace` and `-replaceexpr` and `-by`. The string or expression you specify with `-replace` or `-replaceexpr` is replaced by the string you specify with `-by`. In case of editing, a backup file with a random numeric extension is placed next to the resulting file. E.g. when `index.html` is edited there'll be a file `index.html.1234` or something similar. (DISCLAIMER: the author cannot be held responsible for any damage resulting from using the `edit-` or any other functions of webxref or indeed any software, hardware, chemical substance, imagined or real (or imagined to be real) effects or by-effects of anything, at all, whatsoever.)

```
-find string  report files containing the given string
-findexpr regexp report files containing the given expression
-replace string *REPLACE* string by the string given with -by
-replaceexpr regexp *REPLACE* regexp by the string given with -by
-by string    replacement string (or regexp)
-nobackup     Not implemented on purpose.
```

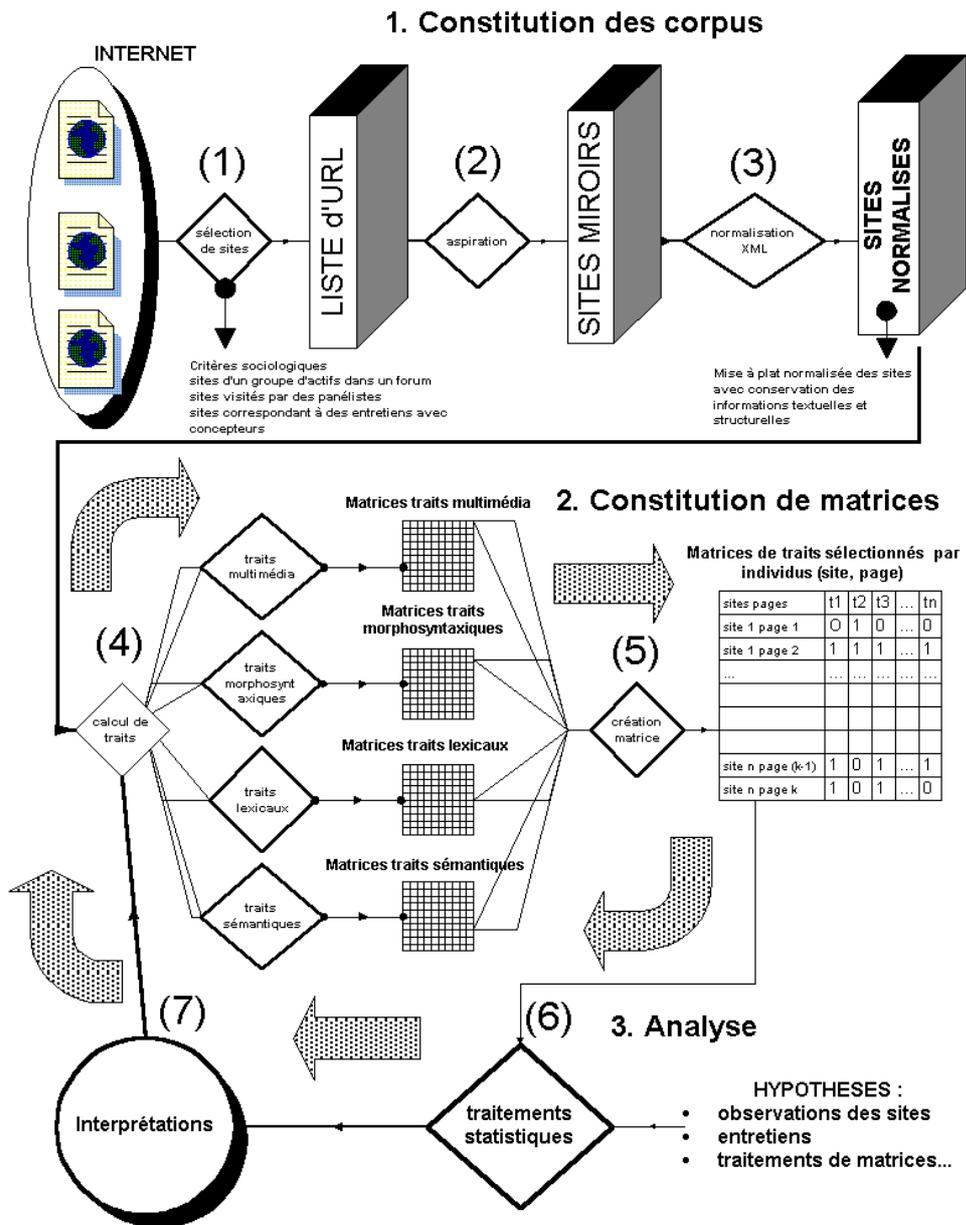
```
=====
Examples
=====
```

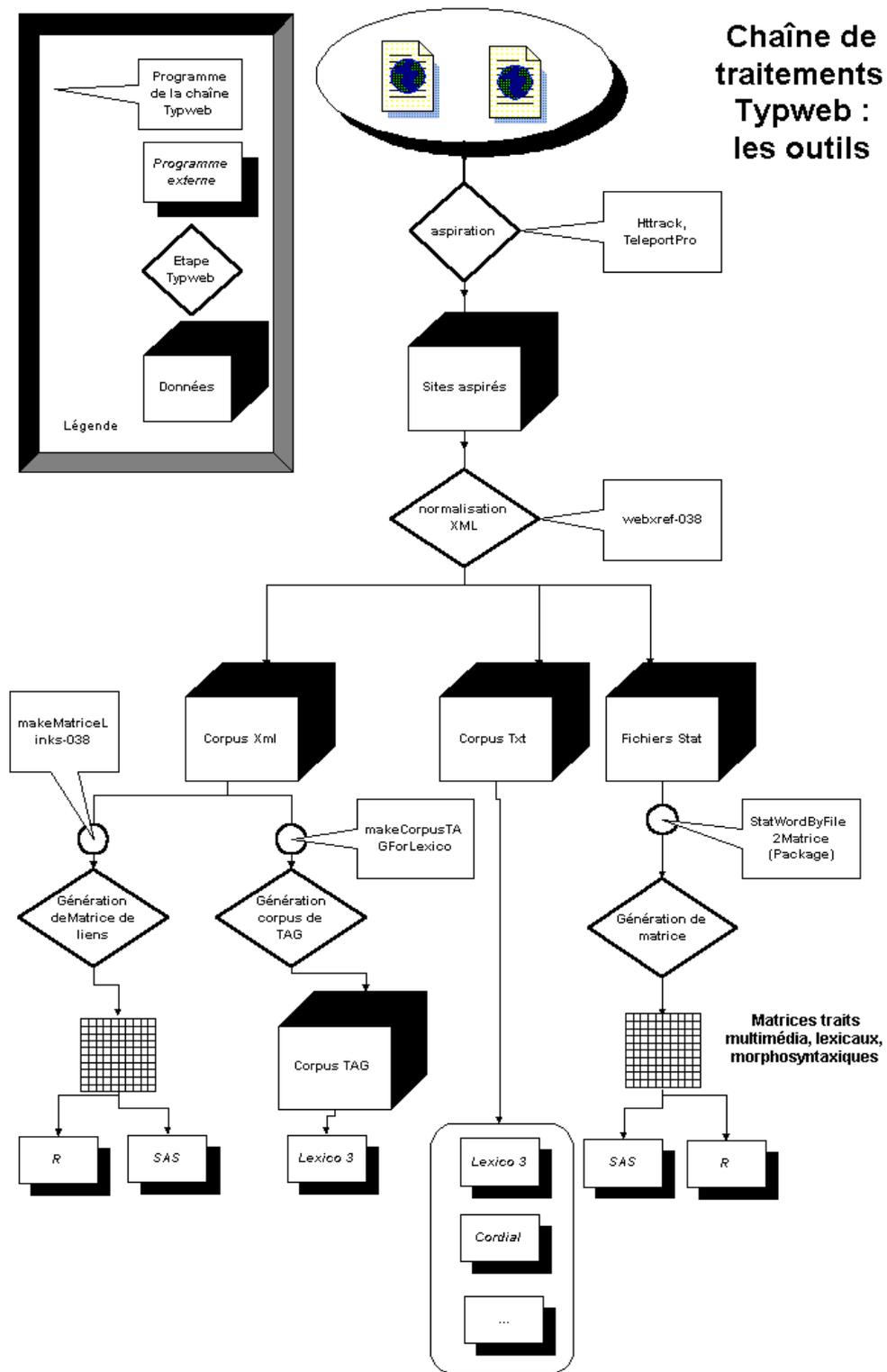
```
webxref file.htm(1) or webxref site/
    Lists every file encountered in directories, reports problems,
    dissects .html, .html and writes the list of the reports in
    "/res(site)[]/analysis_results.html[txt]".
webxref sitel/ site/2
    Analyse site(directory)1 then site(directory)2
webxref -at path file.htm(1)
```

```
Lets you choose a directory on you disk where to put the results
webxref -norep file.html
lists files encountered in directories and reports problems
webxref -html index.html
lets you get the reports in .html form
webxref -one index.html
just check the links in index.html, don't follow the links
webxref -one *.html
Check only the links in the html-files in the current dir.
webxref -depth 0 index.html
Check index.html, but don't check files in directories
that are deeper in the file system.
webxref -http file.html
Checks file.html and external URLs
webxref -htmlonly file.html
Checks file.html, but only files with the .html/htm extension
webxref -avoid '*Archive.*' file.html
Checks file.html but avoids files with names containing
'Archive'
webxref -avoid '*Archive.*|*.Distribution.*' file.html
Same as above, but also files with names containing
webxref -islocal www.sara.nl
Treat things like '<a href=http://www.sara.nl/rick' as a
local reference, as if it would have been '<a href=/rick'

webxref -fluff index.html
Checks index.html and reports files in the directories
encountered that were not referenced by index.html or any
file linked to from there.
webxref -silent index.html
Just report problems at the end of the run. This may take
a while with a big website.
webxref -silent -errors index.html
Prints only problems while scanning, and the final report.
webxref -verbose index.html
Prints a message for every file under test.
webxref -brief -silent index.html
Does not print messages while scanning, and generates a
short report, i.e. lists just problems.
webxref -before -date 970823 -time 1200 index.html
Reports files last modified before August 23rd 1997
webxref -find 'me.gif' index.html
Reports a list of pages containing the text 'me.gif'
webxref -findexpr '<img .*\.gif' index.html
Reports files containing links to gif files.
webxref -replace 'me' -by 'you' -one index.html
Replace 'me' by 'you' in index.html one-ly.
```

6.2 Aide Typweb





S. Fleury
01/07/2001

Les options disponibles dans le menu Typweb :

TOOLS 036

1. Read WebXref Results Site (036) : lancement du programme mktipo (normalisation du site) sur un corpus traité par webxref 036
2. Make stat (036) :
3. Tag HTML (036) :
4. Select Markup (036) :

TOOLS 038

1. TAG-WORD Matrice :

2. LINKS Matrice :
Format d'entrée :

- le programme prend en entrée un corpus XML construit par webxref -038.

Format de sortie :

- le programme cree en sortie un tableau d'un certain nombre de lien (interne, externe (http,ftp), et mailto) référencé dans le corpus XML.

3. TAG corpus :