

Projet L017

Préparation du corpus et indexation

Introduction

Le projet de LO17 consiste en un archivage des pages d'information du site LCI (issues de la rubrique « Monde » et récoltées entre le 25/02/2005 et le 02/03/2006). On souhaite ensuite pouvoir indexer ces données afin de pouvoir par la suite effectuer des requêtes en langage naturel, à la manière d'un moteur de recherche tel que Google, pour obtenir des informations sur les articles à partir de critères tels qu'une date, un thème, un contenu ou un lieu par exemple.

Dans un premier temps, il nous a fallu préparer le corpus documentaire qui a été récupéré sous forme de pages HTML et l'indexer via des fichiers inverses permettant de décrire les documents à l'aide de tout ou partie des éléments contenus dans le corpus.

Préparation du corpus

La phase de préparation du corpus consiste à partir des pages HTML contenant les articles du site LCI à générer un fichier XML structuré de telle manière que l'on puisse référencer l'ensemble des articles selon un squelette commun :

- la une
- les « à voir aussi »
- le focus
- les gros titres
- les rappels (ou brèves).

Vous êtes ici : [Accueil](#) > [News](#) > [MONDE](#) Samedi 19 mars - Mise à jour : 10h21

MONDE

[Irak : deux ans après, les opposants toujours là](#)



[Deux ans jour pour jour après le début de l'intervention américaine en Irak, les opposants à la guerre défilent aujourd'hui à Londres et à New York.](#)

[Aux Etats-Unis, le soutien populaire s'est fortement érodé.](#)

[Lire l'article](#)

A voir aussi :

[Jour historique pour l'Assemblée irakienne \(16/03/2005\)](#)

[Irak : Berlusconi recule \(16/03/2005\)](#)

LE FOCUS

["Une communauté de destin UE-Russie"](#)

[Lors d'une conférence de presse commune à l'Élysée, Chirac, Schröder, Zapatero et Poutine se sont efforcés vendredi soir de parler d'une seule voix sur les grands enjeux internationaux.](#)

[Lire l'article](#)

[Liban - L'appel au dialogue de Lahoud](#)
Le président libanais Emile Lahoud a appelé samedi après un attentat qui a fait 11 blessés, l'opposition et les pro-syriens au dialogue "pour sauvegarder le Liban".

[Trafic - 1,2 million d'enfants vendus dans le monde](#)
Selon l'OSCE, le trafic d'enfants est en progression rapide dans le monde. Il aurait doublé dans la seule Europe du sud-est au cours des trois dernières années. Une criminalité florissante, alors que nombre de pays manquent encore des outils législatifs nécessaires pour protéger les victimes.

[Proche-Orient - Les groupes palestiniens prolongent la trêve](#)

A l'issue de trois jours de discussions en Egypte avec Mahmoud Abbas, les groupes armés palestiniens ont accepté de prolonger l'accalmie actuelle avec Israël. Ils posent néanmoins certaines conditions, notamment la libération de prisonniers.

[Irak - Berlusconi fait marche arrière](#)

Après avoir annoncé un retrait progressif des troupes italiennes d'Irak dès septembre, le chef du gouvernement italien a fait marche arrière mercredi soir, conditionnant ce retrait à un accord avec les Etats-Unis et le Royaume-Uni.

16 mars - Irak - [Jour historique pour l'Assemblée](#)

16 mars - UE - [Porte close pour la Croatie](#)

16 mars - Pédophilie - [500 internautes arrêtés dans 12 pays](#)

15 mars - USA - [Les gavs gagnent une bataille](#)

→ La "une"

→ Les "à voir aussi"

→ Le "focus"

→ Les "gros titres"

→ Les "brèves"

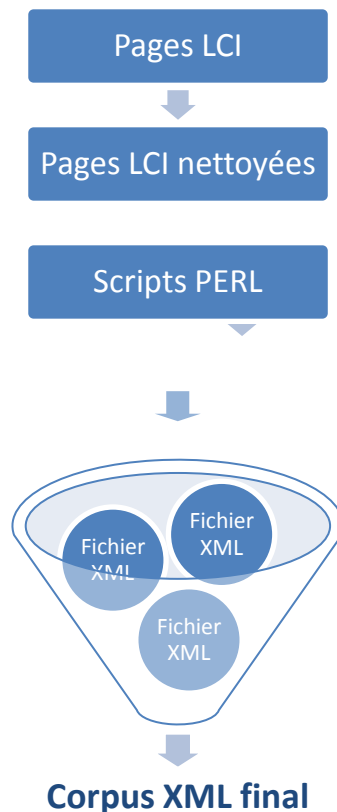
Dans un premier temps, il a été nécessaire de supprimer toute sorte de sauts de ligne afin d'obtenir tous les articles sur une seule et même ligne ; ceci a permis ensuite de faciliter le découpage en rubriques. On en a également profité pour encoder le contenu des articles en ISO-8859-1 (charset latin1). Cet encodage est certainement discutable, dans la mesure où l'encodage standard des fichiers XML est l'UTF8 (unicode), format qui est de plus en plus utilisé et considéré en tant qu'encodage standard car il permet d'utiliser toutes sortes d'accents et de caractères (idéogrammes, ,etc...). De plus la console Linux utilisée pour développer a quelques soucis d'affichage des accents encodés en ISO-8859-1...

Afin de repérer les différentes sections dans une page HTML LCI, il a donc fallu étudier les différents articles et créer des expressions régulières afin de découper le fichier et mettre chaque rubrique (une, voir aussi, focus, gros titres et rappels) sur une ligne qui lui est propre (réalisé en PERL). On a pu s'apercevoir que les blocs des différentes rubriques étaient caractérisés par des balises particulières (ex : *Blc=27303*) qui n'étaient pas toujours les mêmes (des fois 2 balises différentes pour une même rubrique dans 2 articles différents).

Balises séparatrices	Rubrique
<code><!-- Bloc IBL_ID=27914 - "GeneralOuvre" --></code> <code><!-- Blc=27914, "GeneralOuvre" --></code>	La une
<code><!-- Bloc IBL_ID=27913 - "news/NewsAutresArticles" --></code> <code><!-- Blc=27913, "news/NewsAutresArticles" --></code>	Le focus et les « à voir aussi »
<code><!-- Bloc IBL_ID=27915 - "news/NewsAutresArticles" --></code> <code><!-- Blc=27915, "news/NewsAutresArticles" --></code>	Les gros titres
<code><!-- Bloc IBL_ID=27916 - "news/NewsAutresArticles" --></code> <code><!-- Blc=27916, "news/NewsAutresArticles" --></code>	Les brèves

Un petit script shell a été réalisé afin de vérifier pour chaque fichier HTML (contenant chaque rubrique sur une ligne) si toutes les rubriques avaient bien été détectées et traitées (*verif_lignes_all.sh*).

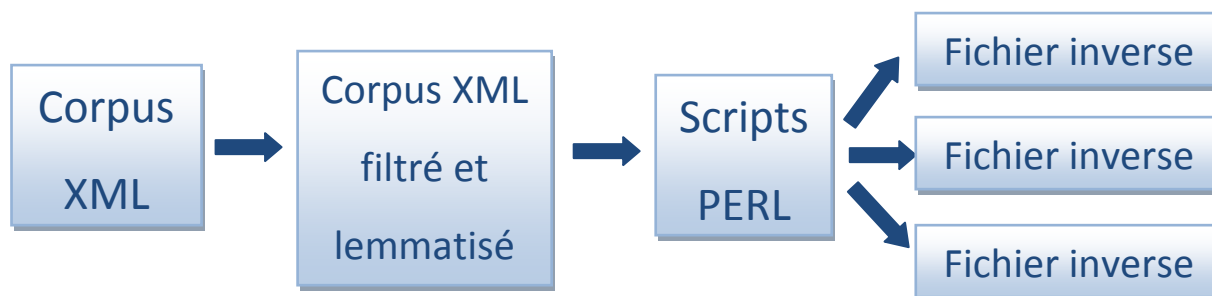
La création de l'arbre XML a été décomposée en plusieurs scripts PERL : le principal génère l'arbre XML à partir d'une page HTML nettoyée comme décrit précédemment (*write_une_page.pl*). Ce script fait appel à plusieurs autres scripts PERL qui s'occupent de générer la structure XML de chaque rubrique (*write_une.pl*, *voir_aussi.pl*, *write_focus.pl*, *write_gros_titres.pl* et *write_rappels.pl*). Chaque structure XML (rubrique) référence l'url de l'article, son titre, sa date, l'url de son image (si présente), son résumé, son thème éventuellement et son auteur (avec son adresse mail si présente).



Afin de vérifier que le fichier XML est bien formé, nous avons créé un petit script PERL qui compte le nombre de pages LCI contenues dans le corpus XML généré afin de vérifier qu'il correspond au nombre de pages HTML du corpus initial (*NB_PAGE_LCI.pl*). De plus, nous avons vérifié « manuellement » que certaines pages HTML du début, du milieu et de la fin du corpus étaient bien retranscrites dans le fichier XML final (sélection d'une partie du corpus aléatoirement et vérification).

Indexation

L'objectif de cette seconde partie est de créer des fichiers inverses à partir du corpus XML créé lors de la première phase. Chaque fichier inverse correspond à une balise (date, thème, auteur) ou un ensemble de balises (titres et résumés). Ceci permettra par la suite de récupérer la liste des pages, rubriques et URL des articles à partir d'un mot entré par l'utilisateur lors de la recherche.



Dans un premier temps, nous avons retiré les petits mots du corpus qui n'étaient pas significatifs (entre 1 et 4 lettres tels que les articles, déterminants, etc.). En effet le fait de saisir un de ces mots dans la barre de recherche ne restreindrait en aucun cas la recherche, ce qui prouve bien leur inutilité.

Après, nous avons recensé les différents petits mots du corpus que l'on a ensuite trié selon leur fréquence décroissante (script `frequence_pt_mots.pl`) et on a supprimé les mots au dessus du premier mot significatif (dans notre cas le mot « Bush » qui apparaît 297 fois dans le corpus). Ceci s'effectue par observation dans le cadre de ce TP, mais il est évident que dans le cas du Web par exemple avec Google, on se base sur les statistiques et les probabilités.

Dans un deuxième temps, nous avons récupéré la liste des mots des titres et résumés des articles et appliqué les scripts `tronc.pl` puis `filtronc.pl` – « algorithmes de troncation et de lemmatisation » – afin de récupérer les lemmes correspondants aux mots du corpus. Cet algorithme consiste d'abord à calculer pour chaque lettre de chaque mot le nombre de lettres différentes que l'on trouve parmi les mots qui ont le même préfixe, ensuite le lemme est déterminé par le premier minimum suivi du plus grand maximum rencontré.

Exemple de résultat du script de troncation :

```

999651210  contrôlé
999651240  contrôle
9996512410 contrôler
9996512410 contrôles
99965124110 contrôleur
  
```

Exemple de résultat de la lemmatisation :

```

contrôlé  contrôl
contrôle  contrôl
contrôler  contrôl
contrôles  contrôl
contrôleur  contrôl
  
```

On voit donc que pour les mots « contrôlé, contrôle, contrôler, contrôles, contrôleur », seul le lemme « contrôl » a été retenu par l'algorithme de lemmatisation, ce qui permet de ne stocker qu'un unique lemme pour les 5 mots de départ (gain en stockage) et permettra par la suite plus de flexibilité sur l'orthographe de ce mot ou l'utilisation de cette racine.

- **Limites de l'algorithme de troncation :**

L'algorithme de troncation nous a aidé à réduire les différentes formes des mots du corpus, par exemple les mots « contrôlé, contrôle, contrôler, contrôles et contrôleur » étaient réduits au lemme « contrôl ». En revanche cet algorithme a aussi des inconvénients qui proviennent de la perte d'informations résultant du remplacement d'un mot par son lemme.

Après la lemmatisation du corpus, il a fallu filtrer les mots c'est-à-dire lister les mots non-significatifs afin de les retirer du corpus et lister les mots avec leurs lemmes correspondants. Il est nécessaire de conserver les petits mots significatifs (en dessous d'un seuil de fréquence comme décrit précédemment. Ex : « Bush »). Il faut également supprimer les mots rares qui n'apportent pas d'information sur un document ni sur le corpus (en dessous d'un certain seuil observé pour ce TP). Ces filtres appliqués au corpus XML généré lors de la première phase permettent de ne conserver que les mots significatifs et de remplacer les mots par leurs lemmes.

Enfin, la création des fichiers inverses s'est déroulée de deux manières différentes :

- Exécution du script `index.pl` pour générer les fichiers inverses correspondants aux balises « `dateArticle` », « `themeArticle` » et « `mailto` ». Ceci permettra par la suite d'effectuer des requêtes sur les dates des articles, leur thème et leur auteur.
- Exécution du script `newindexMot.pl` pour générer les fichiers inverses correspondants aux titres et aux résumés des articles. On a généré trois fichiers dans ce cas de figure : un qui contient les lemmes des titres et des résumés, un qui contient les lemmes des titres uniquement et un dernier qui contient seulement les lemmes des résumés. Ceci permettra au final d'exécuter des requêtes sur les différents mots contenus dans les titres et/ou résumés des articles, à la manière d'un véritable moteur de recherche.

Chaque fichier inverse est en réalité une liste de lemmes suivis de la liste des pages, rubriques et URL des articles dans lesquels ils apparaissent. Leur génération a nécessité la création de fichiers temporaires qui ne référencent qu'une page/rubrique/URL pour chaque lemme, qu'on regroupe ensuite dans un script final afin d'avoir pour chaque lemme la liste des pages qu'il référence.

Exemple :

Lemme	Page	Rubrique	URL de l'article
olympi	/lci-monde-2005-04-26.html	grostitre	news/monde/0,,3215465-vu5wx0leiduy,00.html
	/lci-monde-2005-04-27.html	grostitre	news/monde/0,,3215465-vu5wx0leiduy,00.html

Conclusion

Les deux premières étapes de nettoyage du corpus et d'indexation ont permis de mettre en pratique :

- ❖ L'extraction de données dans un corpus à l'aide d'expressions régulières
- ❖ La génération d'un corpus normalisé et structuré (XML) à partir de nombreux documents dépourvus de structure au premier abord (HTML)
- ❖ Filtrer un corpus afin de retirer les mots non-porteurs d'information, non-signifiants
- ❖ Remplacer les mots du corpus par leurs lemmes équivalents (lemmatisation) afin d'indexer les données de manière optimisée et flexible (correction orthographique par la suite ou suggestions comme dans Google...)

Ainsi, grâce à l'appréhension accrue du langage PERL, nous sommes désormais en mesure d'utiliser ces données indexées dans des tables inverses afin de récupérer des données via des requêtes.

Pour commencer, nous avons réalisé un petit script Perl qui permet de requêter simplement le corpus (requête booléenne) afin de récupérer les documents qui contiennent une liste de mots passée en paramètre (option pour séparer les mots par des « OU » ou des « ET »).

Nous allons donc par la suite améliorer le système de requêtage à l'aide du langage universel SQL et permettre de renvoyer les documents associés dans un ordre particulier, du plus pertinent au moins pertinent (en mode booléen ils sont tous équivalents pour le moment).