Annexe B

Esquisse des algorithmes et structures de données pour la statistique textuelle

Les logiciels décrits ou évoqués dans l'annexe A permettent la segmentation automatique d'un texte, l'indexation des unités textuelles, la recherche des contextes, etc. Les similitudes que l'on note entre ces différents logiciels montrent que leurs auteurs ont trouvé des solutions souvent très proches aux problèmes rencontrés ; les différences qui existent entre ces logiciels témoignent par ailleurs de leurs objectifs distincts.

Il nous a semblé utile de décrire sommairement dans cette annexe les principaux algorithmes qui sont à la base du logiciel *Lexico1*. Pour ce logiciel, les méthodes relatives à la segmentation automatique et à l'indexation des occurrences d'une forme sont fondées sur une structure de données particulièrement simple.[1] La numérisation du texte proposée permet, dans la plupart des cas, de stocker à la fois en mémoire vive le texte numérisé et le dictionnaire des formes. Cette structure s'est révélée très efficace pour la réalisation des principales tâches au sein d'un logiciel dont l'ambition se limite au domaine de la recherche lexicométrique.

Classification des items textuels

Dans ce qui suit, on appellera *item* toutes les occurrences des unités que l'on peut rencontrer lors du dépouillement d'un texte (occurrences de formes graphiques, occurrences de ponctuations diverses, etc.), on appellera ces unités elles-mêmes des *articles*.

L'algorithme de segmentation automatique repose sur une classification des items (et articles) du fichier-texte. Cette classification implique qu'au moment de soumettre le texte au processus de segmentation un certain nombre d'ambiguïtés

[1] Cette structure a pu être améliorée à la suite de discussions avec des spécialistes du domaine de l'informatique textuelle et avec J. Dendien en particulier.

aient été levées et que l'on puisse considérer comme réalisée, par rapport au processus de segmentation, la condition :

un signe de l'enregistrement = un statut

Les items que l'on s'attend à rencontrer appartiennent à deux grandes catégories : les *occurrences de forme* et les *jalons textuels*. Les items de la première catégorie sont les unités dont le décompte motive l'entreprise de la statistique textuelle telle qu'elle a été définie aux deux premiers chapitres.

L'ensemble des jalons textuels se subdivise en deux catégories les *ponctuations* et les *clés*.[1]

Les clés

Les clés permettent d'introduire dans le texte des informations péritextuelles de toutes sortes. Elles sont du type :

<Type1=va11>

Les clés sont introduites entre deux caractères réservés < et >. Le signe "=" sépare le type et le contenu de la clé.

Type1	—	indique le *type* de la clé.
val1	—	qui peut être précédé par des caractères "blancs" ne faisant pas partie de ce contenu, indique un *contenu* de clé, c'est-à-dire une valeur que l'on assigne à cette clé.

Les ponctuations

La classe "ponctuation" rassemble une classe de jalons textuels qui dépasse largement l'ensemble des signes que l'on regroupe sous cette appellation dans le langage courant.[2]

[1] Le système de codage des clés présenté ici s'inspire très largement de celui présenté dans Lafon et al. (1985).

[2] Bien que ce signe ait une fonction complètement distincte au plan de l'encodage du texte, de celle des signes usuels de ponctuation, le caractère "retour chariot" peut être considéré du point de vue de la segmentation automatique comme un caractère de type "ponctuation".

La liste des signes qui seront considérés comme signes délimiteurs de forme par le programme de segmentation est déterminée par l'utilisateur.[1]

A ces signes délimiteurs de formes s'ajoutent obligatoirement[2] :

— les caractères < et > qui servent à introduire les clés
— le caractère "blanc"
— le caractère "RC" ou retour chariot qui sert à découper le flot d'entrée en lignes.

Avec cette convention on peut regrouper en trois classes l'ensemble des jalons textuels que l'on trouve dans un texte :

ponctuation	—	caractère délimiteur de forme (à l'exclusion des signes réservés < et >)
type de clé	—	suite de caractères non-délimiteurs, précédée par le signe < et terminée par le signe "="
contenu de clé	—	suite de caractères non-délimiteurs terminée par le signe >

La segmentation automatique du texte

Le but de l'étape de segmentation automatique du texte est de permettre la construction, à partir d'un fichier texte que l'on appellera *text1*, d'une base textuelle numérisée qui servira de point de départ à l'ensemble des algorithmes de documentation et d'analyse statistique. Cette base est constituée par deux fichiers complémentaires :

text1.dicnum ou *dictionnaire*
et *text1*.textnum ou *fichier texte numérisé*

Le dictionnaire

Le fichier dicnum contient un enregistrement pour chacun des articles du texte à l'exception des articles du type "ponctuation". Les enregistrements correspondant aux articles sont classés en fonction de leur type dans l'ordre suivant :

[1] Le programme Segmentation propose à l'utilisateur une liste contenant les signes de ponctuations les plus courants [. ? ! ; : , " ( ) / ' _ §]. Cette liste est entièrement modifiable par l'utilisateur. Le caractère § peut être utilisé pour marquer le début de chacun des paragraphes du texte de référence.

[2] Ces signes délimiteurs de forme sont rajoutés automatiquement par le programme de segmentation à la liste des caractères délimiteurs choisis par l'utilistateur.

formes	—	interclassées par ordre lexicométrique[1] à l'intérieur de cette catégorie (i. e. par ordre de fréquence décroissante, l'ordre lexicographique départageant les formes de même fréquence).
types de clés	—	classés par ordre lexicographique
contenus de clés	—	classés dans le même ordre

Chacun de ces enregistrements contient les renseignements suivants :

lexicog.	—	ordre lexicographique de l'article dans la liste ci-dessus (l'ordre lexicographique pour les formes).
fréq.	—	la fréquence de l'article
fgraph.	—	la forme graphique de l'article : forme dans le cas d'une occurrence textuelle, suite de caractères donnant le type ou le contenu d'une clé dans les autres cas.

Le nombre des formes différentes est noté *nbform,* celui des articles *nbarticles.*
Le texte numérisé

Le fichier dicnum se présente sous la forme d'une suite de **nbitem** (= nombres des items du texte). Chacun de ces nombres correspond à un des items du texte.

B.1 Tâche n°1 : La numérisation du texte

En fonction de la place mémoire disponible[2] le programme commence par fixer la taille maximale du problème que l'on pourra traiter dans l'environnement considéré.

Les deux principaux paramètres du problème sont *ItemMax* et *ArtMax* respectivement égaux au nombre maximal des items que l'on se propose de traiter et au nombre maximal d'articles que l'on s'attend à trouver dans un texte ItemMax items.

[1] Cf. les définitions du chapitre 2.
[2] A titre indicatif, cette structuration des données permet de traiter environ 700 000 occurrences sur une machine disposant de 4MO de mémoire vive.

On réserve ensuite les tableaux à une dimension **T**(*ItemMax*), **V**(*ArtMax*) ainsi qu'une série de tableaux de travail. L'un de ces tableaux est muni d'une structure d'arbre binaire dont le noeud élémentaire est structuré de la manière suivante :

structure ***tdic*** *{*

mot :	pointeur sur une zone texte qui contient la forme graphique de l'article;
freq	variable entière contenant le nombre des occurrences de l'article rencontrées depuis le début du processus;
lexicog	rang lexicographique de l'article (qui sera calculé après la fin de la lecture du texte);
lexicom	rang lexicometrique de l'article (idem);
numorg	numéro de l'article dans la liste des articles classés par ordre d'apparition dans le texte;
suivant	pointeur qui pointe sur le noeud suivant;

}

Le tableau **Ftex** est un tableau de caractères dans lequel sont mis bout à bout les suites de caractères qui composent tant les forme graphiques que les types et contenus des clés.

L'adresse du premier de ces caractères est stockée dans la variable *mot* de la structure **tdic**. Le dernier caractère de la forme est suivi d'un caractère spécifique qui permet de le repérer en tant que tel.

La deuxième phase de l'algorithme voit une lecture du fichier texte et une première numérisation item par item. Les items qui se présentent dans le flot d'entrée sont isolés et analysés par la procédure **LireItem** qui calcule pour chacun d'eux :

- le numéro d'item dans le texte depuis le début du texte
- le statut de l'item par rapport au 4 catégories d'articles (occurrence, ponctuation, type de clé, contenu de clé)
- la forme graphique de l'item

Chaque item est ensuite présenté à la racine de l'arbre binaire (ou d'un B-arbre) pour être comparé aux articles déjà stockés dans l'arbre. Les articles sont comparés sur une base lexicographique (aa < ab).

Ce processus permet de calculer un code numérique *CodNum* pour chacun des items entrés. Si l'item considéré correspond à l'occurrence d'un article qui est déjà apparu lors de l'exploitation en cours, sa présentation à l'arbre de stockage permet de retrouver et d'incrémenter la variable qui comptabilise le nombre de ses

occurrences. Dans le cas contraire, il s'agit d'un nouvel article dont le numéro de stockage est calculé par incrémentation de la variable qui comptabilise le nombre des objets stockés dans l'arbre.

A la fin de la lecture du fichier texte, l'arbre contient tous les articles présents dans le texte. Pour chaque article on connaît en outre le nombre de ses occurrences dans le texte. Les articles sont numérotés d'après l'ordre de leur apparition dans le texte. On en profite pour affecter ce numéro d'ordre à la variable *numorg*.

Les tris

Le calcul des rangs lexicographiques et lexicométriques de chacun des articles est réalisé à partir du fichier des articles.[1] On commence par remplir *List (*)* un tableau de pointeurs de longueur *nbarticles* dont chacun pointe sur un des noeuds de l'arbre de stockage des articles.

Ce tableau de pointeurs est trié une première fois d'après l'ordre lexicographique de chacun des articles.[2] Les articles correspondant aux formes graphiques se trouvent placés en tête de cette liste du fait de l'adjonction de caractères de poids très élevé devant les chaînes de caractères correspondant aux autres catégories. Les articles du type "type de clé" sont suivis eux-même par les articles du type "contenu de clé". Après cette opération, on peut affecter à la variable *lexicog* une valeur égale au rang de l'article après tri de l'ensemble selon l'ordre lexicographique.

Un second tri par ordre décroissant des pointeurs du tableau *List (*)* d'après la valeur de la variable *freq* de l'article sur lequel ils pointent, les formes de même fréquence étant départagées d'après les valeurs ascendantes de la variable *lexicog* que nous venons de calculer, permet de classer les articles correspondant aux formes graphiques d'après l'ordre lexicométrique. Cette opération terminée, on peut assigner à la variable *lexicog* le numéro d'ordre lexicométrique.

Il est possible alors de passer à l'écriture sur support externe du dictionnaire. Cette écriture se fait en suivant l'ordre lexicométrique dans lequel sont actuellement triés les pointeurs contenus dans le tableau list. Pour chaque article on écrit successivement :

[1] Cette manière de procéder constitue un progrès important par rapport aux algorithmes de segmentation qui effectuent les tris sur le fichier des items beaucoup plus volumineux que celui des articles.

[2] Dans la pratique, ce tri est notablement accéléré par une technique de "tri par morceaux". Les articles sont d'abord regroupés en sous-groupes en fonction du premier caractère. On effectue ensuite un tri à l'intérieur de chacun des sous-groupes.

freq	le nombre des occurrences de l'article dans le texte;
lexicog	rang lexicographique de l'article;
mot	suite de caractères composant l'article terminée par un retour chariot.

Un dernier tri de la liste de pointeurs contenue dans le tableau list, effectué d'après les valeurs de la variable *numorg,* replace cette liste dans l'ordre initial.

La numérisation finale

La dernière phase de l'algorithme de segmentation permet de stocker sur support externe le texte numérisé d'après l'ordre lexicométrique des articles du texte. Pour effectuer cette opération il suffit de reprendre le tableau T dans lequel les articles sont numérisés d'après l'ordre de leur apparition dans le texte et de substituer le numéro lexicométrique de chaque article à ce numéro. On trouve le numéro lexicométrique correspondant à l'article dont le numéro d'apparition est égal à i dans la variable *lexicom* du noeud de l'arbre de stockage pointé par l'élément i.

B.2 Tâche n°2 : La recherche de contextes

On regroupe sous cette appellation la recherche de différents sites textuels pour un ensemble d'occurrences correspondant à une *forme-pôle* donnée ou à une liste de ces formes. On construit une telle liste en introduisant l'une des sélections suivantes :

- une forme graphique
- une liste de formes graphiques
- un groupe de caractères constituant le début d'une ou de plusieurs formes présentes dans le texte.
- un groupe de caractères constituant la fin d'une ou de plusieurs formes présentes dans le texte.
- un groupe de caractères présent dans le corps d'une ou de plusieurs formes présentes dans le texte.

Les unités de contexte retenues pour l'ensemble des formes-pôle sélectionnées peuvent être :

- une ligne de contexte

— un nombre fixe de lignes de contexte pour chacune des occurrences de la liste des formes-pôle.
— un inventaire distributionnel des segments répétés après la forme-pôle.

Les contextes peuvent-être triés selon deux types d'arguments dont chacun peut être choisi comme argument majeur du tri ou comme argument mineur :

— en fonction des formes graphiques
— en fonction des valeurs d'une clé sélectionnée dans le texte

Pour une forme graphique donnée, les contextes peuvent être également triés, par rapport à la forme-pôle :

— en fonction de l'ordre lexicographique des formes qui précèdent chacune des occurrences de cette forme.
— en fonction de l'ordre lexicographique des formes qui suivent chacune des occurrences de cette forme.
— en fonction de l'ordre d'apparition des occurrences de cette forme dans le texte.

B.3 Tâche n°3 : Le calcul de la gamme des fréquences.

Le calcul de la gamme des fréquences peut être réalisé à partir du seul dictionnaire dans lequel les formes qui ont une même fréquence sont classées côte à côte. En commençant par la fréquence la plus élevée (ou au contraire par la fréquence 1) on calcule par simple cumul les effectifs V_i qui correspondent à chacune des fréquences pour i variant de 1 à fmax.

B.4 Tâche n°4 : La construction des Tableaux Lexicaux.

Le tableau lexical est déterminé par deux paramètres fixés par l'utilisateur :

— *fmin* : le seuil minimal retenu pour la sélection des formes qui correspondront aux lignes du tableau lexical (on ne retiendra que les formes dont la fréquence est égale ou supérieure à ce seuil).

— *Sxx* : un type correspondant à une clé (le tableau lexical comptera une colonne pour chacune des valeurs différentes prises par le contenu de ce type de clé dans le fichier texte).

Une fois ces valeurs fixées, on peut calculer *x1*, le nombre des lignes du tableau lexical qui est égal au nombre des formes dont la fréquence est au moins égale à *fmin* dans le corpus. Remarquons que, par suite de la définition de l'ordre lexicométrique que nous avons adoptée, les numéros lexicométriques de toutes les formes dont la fréquence est inférieure à ce seuil sont supérieurs au numéro lexicométrique de la dernière forme de fréquence est égale au seuil retenu. Appelons ce numéro *fder*.

Une exploration des contenus existants dans le texte pour la clé sélectionnée permet de fixer *x2,* le nombre des colonnes du tableau égal au nombre des valeurs différentes prise par la variable "contenu de la clé sélectionnée". Les différentes valeurs prises par cette variable sont triées par ordre lexicographique ascendant et numérotées dans cet ordre. Cette numérotation permet d'affecter à chaque code un numéro de partie.

On peut alors réserver une zone mémoire de taille (*x1* x *x2*) qui contiendra le tableau que l'on se propose de calculer. Le calcul de ce tableau s'effectue en une seule lecture du fichier texte numérisé. En parcourant ce tableau à partir de la première de ses cases on commence par trouver la première occurrence de la clé sélectionnée et le premier contenu de cette clé. Ce contenu nous renvoie à un numéro de partie d'après la table établie précédemment. Ce numéro s'appellera le "numéro de partie courant".

La poursuite de ce traitement pour chacune des cases du tableau "tnum" amène à trois situations différentes :

— le code tnum[k] est :

  soit le code d'une forme dont la fréquence est inférieure au seuil retenu (i.e. un code supérieur à *fder*) ;

  soit le code d'une ponctuation ; soit le code d'un type ou d'un contenu de clé correspondant à une autre clé que la clé sélectionnée et dans ce cas on ignore la case tnum[k] pour passer au traitement du code contenu dans la case tnum[k+1].

— le code tnum[k] est le code de la clé sélectionnée. On lit alors dans la case tnum[k+1] la valeur du contenu de la clé qui permet de mettre à jour le code de partie courant (puisque l'on passe au traitement des occurrences situées sans une autre partie du texte).

— le code tnum[k] est le code d'une occurrence de la forme *forme* dont la fréquence est supérieure ou égale au seuil retenu (i.e. un code inférieur à *fder*). Si ce code est égal à i, on ajoute une unité à la case [i, numéro de partie courant] du tableau lexical que l'on est en train de calculer.

0	de	31	44	52	112	34	72	136	36	48
1	l	35	36	57	82	29	51	82	27	30
2	la	22	28	42	82	27	43	118	25	24
3	les	24	24	32	48	21	33	65	13	26
4	le	36	20	33	61	13	28	71	8	14
5	d	18	20	25	47	14	26	55	7	16
6	pas	15	11	21	43	10	18	72	9	11
7	avenir	21	22	31	34	12	30	28	12	8
8	chômage	21	18	13	35	5	9	58	6	10

Tableau B.1
Exemple de tableau lexical

Lorsque ce processus est achevé on a calculé le tableau lexical des formes de fréquence supérieure ou égale au seuil fixé. On peut éditer ce tableau en faisant précéder la ventilation de chacune des formes sélectionnées par son numéro lexicométrique et son libellé, que l'on trouvera dans le dictionnaire, comme dans le tableau B.1 ci-dessus.

B.5 Tâche n°5 : Tableaux des segments répétés

Les calculs portant sur la ventilation des segments répétés dans les parties d'un corpus de textes permettent de compléter les analyses effectuées à partir des tableaux de formes graphiques (tableaux lexicaux).[1]

Cependant les segments répétés d'un texte sont avant tout caractérisés par leur énorme redondance.

Pour réduire le volume des segments répétés, on écarte de la liste des segments ceux d'entre eux qui sont des segments contraints (i.e. qui sont toujours précédés ou suivis par une même forme et qui entrent donc dans la composition de segments plus longs).

Pour certaines applications statistiques, il est en outre possible de ne considérer que les segments dont la fréquence dépasse un certain seuil de répétition (on abaissera ce seuil à 2 occurrences si l'on désire obtenir la liste de tous les segments répétés dans le texte).

[1] Cf. chapitres 2 et 5.

On peut distinguer pour la réalisation de la tâche n°5 deux sous-tâches, qui interviennent alternativement lors de la construction du TSR d'un corpus.

Ces sous-tâches sont :

5-1 – le repérage des segments non contraints dans le corpus qui commencent par une forme graphique donnée et dont la fréquence dépasse un seuil fixé.

5-2 – le calcul de la ventilation dans les parties du corpus de l'ensemble des occurrences d'un segment ainsi repéré.

Le repérage de l'ensemble des segments non-contraints débutant par une forme donnée, pour chacune des formes graphiques dont la fréquence est égale ou supérieure à un seuil fixé réalise l'ensemble de la tâche n°5.

B.5.1 Sous-tâche 5.1 : Repérage des segments répétés non contraints

Pour une forme graphique donnée *Form1* et un seuil de fréquence *Sfr* fixé, le but de la sous tâche 5.1 consiste donc dans le repérage de l'ensemble des segments non-contraints débutant par *Form1* dont la fréquence est au moins égale à *Sfr* occurrences.
Ainsi définie, cette sous tâche devient très simple à réaliser. On commence par définir une liste de pointeurs list1[k] qui pointent chacun sur une des occurrences de la forme *Form1*.[1]

Dans un deuxième temps ces pointeurs sont triés en fonction de l'ordre lexicographique des formes qui suivent la forme *Form1*.[2]

A la fin de ce tri les segments répétés éventuels qui commencent par la forme *Form1* se trouvent placés les uns à la suite des autres dans l'ordre lexicométrique des segments répétés (i.e. les segments sont classés en fonction de l'ordre lexicographique de la première forme, départagés en cas d'égalité par l'ordre lexicographique de la forme qui suit et ainsi de suite) comme c'est le cas dans l'exemple ci-dessous.[3]

A B

[1] La taille de cette liste est égale, au maximum, à *fmax* la fréquence de la forme la plus fréquente.
[2] Remarquons que cet ordre correspond exactement à celui dans lequel apparaîssent les contextes-lignes lors de la réalisation d'une concordance triée sur le contexte qui suit.
[3] Comme dans les exemples du chapitre 2 relatifs au corpus **P**, les lettres capitales symbolisent ici chacune une forme graphique.

A B
A B C
A B C A C
A B C A C D
A B C A C D
A C B A

Dans le processus de comparaison des segments en vue de compter les occurrences de segments répétés, on procède par comparaison des formes graphiques situées à chacune des positions du segment.

Si le tri effectué nous assure que les segments répétés identiques se trouvent bien côte à côte, il reste à repérer, en parcourant la liste des segments commençant par la forme *Form1*, les segments qui marquent la fin d'un groupe d'occurrences relatives à un même segment de longueur *L1*.

Remarquons qu'à l'intérieur des occurrences des segments commençant par *L* formes identiques, les occurrences d'un segment répété comportant *L+1* formes identiques constituent un ensemble de segments consécutifs après le tri lexicographique des segments.

Comparaison de segments

Les cases des tableaux SEGn[k] sont remplies par des valeurs relatives à des occurrences de la forme *Form1*. Pour le repérage des segments répétés dont la fréquence est supérieure ou égale à *Sfr,* on ne s'intéresse qu'aux séquences ne comprenant aucune forme de fréquence inférieure à ce seuil ne chevauchant pas en outre de délimiteurs de séquences.

Les séquences à comparer sont donc constituées par des suites de codes correspondant au numéro lexicométrique de formes de fréquence supérieure au seuil retenu. On sélectionne un code particulier noté *VIDE* pour remplir les autres cases du tableau SEGn[k].

Par rapport à une occurrence de forme graphique, on appellera dans ce qui suit **forme suivante**, soit la forme *VIDE,* soit la première forme graphique située après cette forme, en négligeant les jalons textuels et signes de ponctuation non-délimiteurs de séquence situés entre les deux formes. Les mêmes conventions s'appliquent pour la définition de la **forme précédente**.

Les séquences débutant par la forme *Form1* qui vont être soumises à la comparaison sont rangées dans les tableaux SEGn[k] de la manière suivante: on range dans la case SEGn[0] le numéro lexicométrique de la forme précédant *Form1*, en respectant les conventions ci-dessus.

La case SEGn[1] contient le numéro lexicométrique de la forme *Form1* qui sert de pivot à la recherche des segments.

Les cases suivantes SEGn[2], SEGn[3],..., SEGn[longmax] contiennent soit les numéros lexicométriques des formes qui suivent soit le code *VIDE* (à partir d'une certaine position).

Recherche de l'accident

La comparaison entre deux segments successifs permet de déterminer la longueur du sous-segment répété le plus long qu'ils ont en commun, c'est-à-dire le nombre des premières occurrences qui sont identiques pour les deux segments.

Nous appellerons *accident* la position pour laquelle une occurrence du second segment ne correspond pas, pour la première fois à l'occurrence du segment précédent. Ainsi, par exemple, dans la comparaison des deux segments

```
      1 2 3 4 5 6 7
      A B C D E F G
et    A B C D E G H
```

nous dirons que l'accident se situe en position numéro 6.

Recherche des segments répétés

Le tableau SEGc[n], qui contient le "segment courant", est initialisé par les valeurs du segment placé en tête par le tri lexicographique des segments. On gère parallèlement un tableau FreqSeg[n] qui contient pour n variant de 2 à *longmax* la fréquence courante des segments de longueur 2 à *longmax*.

On considère ensuite le segment placé immédiatement après par le tri lexicographique. La détermination d'un accident en position *x* par rapport au segment précédent indique que l'on en a terminé avec les occurrences du sous-segment précédent (dont la longueur est égale à *x*). Si la fréquence des segments répétés ainsi répertoriés dépasse le seuil *Sfr,* nous avons repéré un segment répété dont il nous reste à calculer la ventilation dans les parties du corpus.

B.5.2 Sous-tâche 5.2 : Calcul de la ventilation des segments répétés

Le calcul de la ventilation des occurrences du segment répété s'opère à partir du numéro de chacune des occurrences de la forme *Form1*. Ce problème suppose que l'on ait un moyen de déterminer le numéro de la partie à laquelle appartient une occurrence du texte. Plusieurs possibilités s'offrent pour réaliser cette tâche dont la

plus simple consiste à construire un tableau récapitulatif indiquant les cases du tableau *tnum* qui correspondent à des changements de partie.