Une méthode indépendante des langues pour indexer les documents de l'internet par extraction de termes de structure contrôlée

Jacques Vergne

GREYC, Université de Caen, 14032 Caen Cedex

Jacques.Vergne@info.unicaen.fr

http://www.info.unicaen.fr/~jvergne

Résumé :

Nous présentons dans cet article une méthode d'indexation automatique de documents de l'internet, fondée sur l'extraction de termes de structure contrôlée, et qui ne nécessite aucun traitement linguistique, ni stop-list, ni connaissance de la(les) langue(s) du document. Cette méthode s'appuie sur la récurrence de suites de mots, et sur le contrôle de la structure de ces suites. Ce contrôle de structure est basé sur un étiquetage du texte à indexer avec un jeu de deux étiquettes : mots informatifs ou non informatifs. Les mots informatifs sont définis comme étant plus longs et moins fréquents que leurs voisins. On exploite ainsi des propriétés très générales des langues, découvertes par Zipf et par Saussure.

MOTS-CLES : indexation automatique, termes de structure contrôlée, méthode d'indexation indépendante des langues.

Abstract :

In this paper, we present an automatic indexing method of web documents, based on structure controlled terms extraction, and which does not require any linguistic processing, neither stop-list, nor knowing the document language(s). This method relies on the words sequences recurrence, and on the structure control of these sequences. This structure control is based on tagging the text to index with a two label tagset : informative words or not. Informative words are defined longer and less frequent than their neighbours. Very general linguistic properties, discovered by Zipf and by Saussure are thus exploited.

KEYWORDS : automatic indexing, structure controlled terms, language independent indexing method.

1. Introduction

On peut caractériser les méthodes d'indexation automatique actuelles entre deux pôles : d'une part, l'indexation par des termes extraits du document, monolingue, qui nécessite des traitements et des ressources linguistiques (une indexation «en profondeur», issue des pratiques des bibliothécaires), d'autre part, l'indexation par tous les mots du document, dite «full text», utilisée par les moteurs de recherche sur l'internet, consistant en des traitements superficiels de masses textuelles énormes, considérées comme des chaînes de caractères (une indexation «en largeur»), et non pas comme du matériau linguistique.

Ces deux pôles illustrent la divergence croissante des deux problématiques, qui ont paru un temps être analogues : extraction terminologique en profondeur en espace clos et figé, et indexation «full text» en vaste espace ouvert et évolutif. Certains concepts issus de la recherche d'information dans des bases de données textuelles (donc des petits espaces clos, souvent monothématiques et monolingues) ne semblent plus être adéquats à de vastes espaces ouverts. Le silence, par exemple, n'est pas mesurable sur l'internet, car on ne peut pas compter le nombre total de documents pertinents non récupérés. De manière analogue, l'idf (Inverse Document Frequency) garde-t-il son intérêt sur l'internet ?

Nous allons caractériser ces deux pôles, pour introduire ensuite notre méthode.

1.1. L'indexation «full text»

L'indexation «full text» a les caractéristiques suivantes (cf. "text indexing" dans [Salton 83], et dans [Brin & Page 98], l'article fondamental sur Google par leurs concepteurs, écrit avant la création de leur société) :

- tout mot du document est terme et l'indexera, quel qu'il soit (mot plein ou mot vide), quel que soit son effectif (y compris les hapax);
- il n'y a pas de contrôle de structure du terme, car 1 terme = 1 mot ;
- le grain traité est le mot (ou le caractère pour les langues non alphabétiques) dans le document (le mot sera une clé d'accès au document) ;
- les cadres d'utilisation sont les systèmes d'indexation des moteurs de recherche généralistes sur l'internet ;
- le corpus à indexer est constitué d'un très grand nombre de petits documents ;
- le traitement est indépendant des langues (aucun traitement linguistique, seulement reconnaissance de l'écriture alphabétique ou non) ;
- les recherches par terme (expression entre guillemets) se font grâce à l'offset des mots dans les documents (cf. [Brin & Page 98]);
- avantages : aucun traitement linguistique, traitement superficiel, indépendant des langues, permettant d'indexer d'énormes masses textuelles ;
- inconvénients : traitement trop superficiel? trop grand nombre de documents récupérés, dont seulement les premiers classés sont visibles par l'utilisateur; la taille des index devient prohibitive, et nécessitera bientôt une évolution de stratégie

d'indexation : pour chaque mot de chaque document indexé (8.10^9 selon Google[1]), sa graphie est présente une fois dans le dictionnaire de l'index, mais les informations propres à l'occurrence (offset, taille relative de fonte, relation à la graphie, relation à l'identifiant du document) sont présentes autant de fois que d'occurrences.

1.2. L'extraction terminologique de corpus clos

L'extraction terminologique de corpus clos se situe à l'opposé (cf. [Bourigault 02]) :
- un terme est constitué de certains groupes de mots contigus (1 ou plus) du document ;
- la structure syntaxique du terme est contrôlée (ce sont surtout des syntagmes nominaux) ; ce contrôle de structure syntaxique utilise une analyse morphosyntaxique, un dictionnaire, une grammaire monolingues ;
- le cadre d'utilisation est l'extraction de termes structurés (les différentes expansions d'une même tête) d'un corpus clos, monolingue (langue unique identifiée), souvent monothématique ;
- avantage : contrôle très fin de la structure des termes ;
- inconvénient : traitement linguistique lourd, monolingue, nécessitant de reconnaître la langue.

Dans certains systèmes, on exploite uniquement la récurrence en excluant les mots vides à l'aide de stop-lists (cf. [Salem 87], [Salton 93], [Ahonen 99]); si l'on a dans un document : *président de la république* et *président de la société X*, le segment répété est : *président de la* qui n'est pas un terme correct ; ce terme est alors corrigé en en supprimant les mots de la stop-list situés en début ou fin du segment.

1.3. Notre proposition : allier des avantages des deux stratégies

Nous proposons de calculer des termes de structure contrôlée pour indexer des documents (ou des sites) en très grand nombre, de langue(s) inconnue(s), c'est-à-dire d'importer dans l'indexation sur l'internet la finesse du contrôle de structure des termes de l'extraction terminologique de corpus clos.

Le traitement doit être indépendant des langues des documents et donc ne pas utiliser de stop-list.

Définissons les termes comme étant certains groupes de mots contigus, non hapax, et de structure contrôlée : les termes sont centrés sur des mots «informatifs».

2. Principes et algorithmes

On suppose que l'on est dans le cadre de l'indexation d'un document du web (ou d'un site web). Le texte à indexer est extrait du source html du document (ou de certains documents du site, par exemple jusqu'à une certaine profondeur à partir d'un point d'entrée). Pour localiser la partie informative du document (c'est la partie à indexer),

[1] Mais voir "Web: Le mystère des pages manquantes de Google résolu ?" sur le blog de Jean Véronis : http://aixtal.blogspot.com/2005/02/web-le-mystre-des-pages-manquantes-de.html.

nous utilisons la propriété suivante : la partie informative du document est présente une seul fois sur le site[2]. Le texte est découpé en paragraphes (unités physiques marquées dans le balisage), qui constituent l'unité traitée. Dans un document html, les balises de fin de paragraphe sont considérées comme une macro-ponctuation : `title`, `div`, `br`, `p`, `td`.
L'algorithme comporte ensuite 4 étapes :
-1- étiquetage du document avec un jeu de deux étiquettes : mots informatifs ou non informatifs (en première approche, les mots non informatifs sont les mots vides) ;
-2- génération de candidats termes de structure contrôlée ; la structure d'un candidat terme est contrôlée par des motifs fondés sur l'étiquetage mots **I**nformatifs (`I`) ou **n**on (`n`): un candidat terme est présent au moins 2 fois dans le grain à indexer (document ou site), il commence et se termine par un mot informatif, et ne contient pas de ponctuation ; les termes hapax sont supprimés au fur et à mesure du déroulement de l'algorithme ;
-3- élagage de l'ensemble des termes : les termes inclus dans des termes de même effectif sont supprimés ;
-4- pondération des termes dans le document, par une estimation de la place occupée dans le rendu du document, soit par exemple pour un terme : effectif * longueur.

2.1. Étiquetage du document avec un jeu de deux étiquettes

Comment différencier les mots informatifs des mots non informatifs ?
Deux définitions sont possibles :
- la définition classique : les mots informatifs sont les mots pleins ou lexicaux (content words), et les mots non informatifs sont les mots vides, ou grammaticaux (function words); cette distinction a été introduite par [Tesnière 59], page 53; cela conduit alors à l'utilisation de stop-lists, qui ont l'avantage d'être facilement disponibles, mais qui ont deux désavantages : la(les) langue(s) du document doivent être reconnues pour choisir la(les) bonne(s) stop-list(s), et les mots de la stop-list ne sont jamais indexés, par définition, ce qui entraîne qu'un document sur *la vente de stocks d'or* sera inaccessible par la requête «*or*» ;
- nous proposons la nouvelle définition suivante : un mot informatif est plus long et moins fréquent que ses voisins.
Cette définition allie les propriétés linguistiques très générales mise en évidence par Zipf et par Saussure :
- **Zipf** : ce qui est d'usage fréquent est court : c'est la loi de l'économie d'effort dans l'usage d'un code, caractérisée par Zipf [Zipf 49], et observable aussi dans les langages de programmation (remarquons que la «loi de Zipf», toujours très présente dans la littérature, est une loi sur les effectifs des mots, et que les propriétés statistiques des longueurs des mots sont plus rarement invoquées) ;

---

[2] Les algorithmes de localisation de la partie informative d'un document html seront publiés dans un prochain article : relevé de la mise en forme matérielle (MFM), segmentation en paragraphes (unités physiques marquées dans le balisage), calcul des classes d'équivalence de MFM, et calcul de la structure du document autour du corps de texte, tous ces calculs étant conçus pour fonctionner sur de l'html mal formé.

- **Saussure** : *dans la langue, il n'y a que des différences* (cf. [Saussure 22], éd. 1974 p.166); cette importante propriété du matériau linguistique nous conduit à fonder nos calculs sur des différences entre unités, plutôt que sur des valeurs absolues des unités, comme celles que l'on stocke dans les dictionnaires.

Nous proposons d'allier ces deux propriétés dans un calcul complètement local, fondé sur les différences de longueur et d'effectif de deux mots contigus. Dans des travaux antérieurs (cf. [Vergne 03] et [Vergne 04]), à partir des mêmes propriétés, nous avions proposé un algorithme différent, fondé sur le pavage déterministe du segment, avec des motifs de 5 à 2 mots, testés successivement dans un ordre heuristique imposé (`InnI`, `InI`, par exemple). Le nouvel algorithme présenté ici est plus épuré, car il ne fait aucune hypothèse de motif avant l'étiquetage, mais il calcule l'étiquetage uniquement à partir des différences d'effectif et de longueur entre 2 mots contigus.

Dans un paragraphe, soit deux mots contigus (numérotés 1 et 2) d'**ef**fectifs `f1` et `f2` (dans le document ou dans le site, selon que l'on veut indexer chaque document ou le site entier) et de **l**ongueurs `l1` et `l2`.
La différence entre deux mots contigus est définie comme un objet à 2 attributs :
- le type de la différence (la différence est orientée) :
  `si f1>f2 & l1<l2, alors type_diff = nI` (**n**on informatif - **I**nformatif)
  `sinon si f1<f2 & l1>l2, alors type_diff = In` (**I**nform. - **n**on informatif)
  `sinon type_diff = contradictoire` (contradiction entre les deux critères)
- la mesure de la différence :
  `mesure = (max(f1, f2)/min(f1, f2)) * (max(l1, l2)/min(l1, l2))`

Cette mesure permet d'évaluer si la différence est suffisante pour provoquer un changement d'étiquette. Elle est égale au produit des rapports des longueurs et des effectifs. Elle est donc indépendante de la taille du document.

Règle d'affectation d'une étiquette à un mot à partir des différences avec son voisin précédent : si la différence entre les mots 1 et 2 est de type non contradictoire et de mesure suffisante (supérieure à un seuil calculé), alors le mot 2 prend l'étiquette 2 du type de la différence, sinon le mot 2 prend l'étiquette du mot 1, ce qui donne :

```
si type_diff ≠ contradictoire & mesure > seuil,
alors étiq_mot2 = étiquette 2 du type_diff
      si étiq_mot1 indéterminée,
          alors étiq_mot1 = étiquette 1 du type_diff
sinon étiq_mot2 = étiq_mot1
```

Par exemple, voici le déroulement de l'étiquetage de quelques mots :

	f	l	(f = effectif, l = longueur)
1	649	3	*une*
2	32	8	*nouvelle*
3	1	10	*résolution*
4	3673	2	*de*
5	1500	2	*l'*
6	9	3	*ONU*

Pour expliquer l'algorithme, prenons arbitrairement un seuil de 4 (soit un rapport de 2 sur chaque critère) :

différence entre les mots 1 et 2 (*une nouvelle*) :

```
f1>f2 & l1<l2 => type_diff = nI
mesure_diff = (649/32) * (8/3) = 54,1
54,1 > 4 => étiq_mot1 = étiq1_type_diff = n
            étiq_mot2 = étiq2_type_diff = I
```

différence entre les mots 2 et 3 (*nouvelle résolution*) :

```
f2>f3 & l2<l3 => type_diff = nI
mesure_diff = (32/1) * (10/8) = 40
40 > 4 => étiq_mot3 = étiq2_type_diff = I
```

différence entre les mots 3 et 4 (*résolution de*) :

```
f3<f4 & l3>l4 => type_diff = In
mesure_diff = (3673/1) * (10/2) = 18365
18365 > 4 =>  étiq_mot4 = étiq2_type_diff = n
```

différence entre les mots 4 et 5 (*de l'*) :

```
f4>f5 & l4≤l5 => type_diff = nI
```

(égalité possible sur un des 2 critères)

```
mesure_diff = (3673/1500) * (2/2) = 2,45
2,45 < 4 =>  étiq_mot5 = étiq_mot4 = n
```

(car différence insuffisante)

différence entre les mots 5 et 6 (*l'ONU*) :

```
f5>f6 & l5<l6 => type_diff = nI
mesure_diff = (1500/9) * (3/2) = 250
250 > 4 => étiq_mot5 = étiq2_type_diff = I
```

d'où l'étiquetage :

<n>*une* <I>***nouvelle*** <I>***résolution*** <n>*de* <n>*l'*<I>***ONU***

Comment choisir la valeur du seuil de mesure de différence ? Nous proposons que ce seuil soit calculé pour chaque paragraphe.

Dans l'exemple précédent, l'étiquette d'un mot a été affectée en fonction de sa différence avec le mot précédent. On peut aussi calculer son étiquette en fonction de sa différence avec le mot suivant. D'où une mesure de la qualité d'un étiquetage en fonction d'une valeur de seuil : le nombre de désaccords d'étiquetage entre les étiquetages selon la différence avec le mot précédent ou bien avec le mot suivant.

On observe que la fonction : nombre de désaccords d'étiquetage = f(seuil) a un minimum, qui permet de définir la meilleure valeur du seuil.

D'où l'algorithme de calcul du seuil qui consiste à rechercher un minimum approché de cette fonction entre 2 bornes, de valeurs initiales :

`borne basse = 1` (égalité sur les 2 critères pour les 2 mots)

`borne haute = racine carrée (f_max * l_max) / 2`

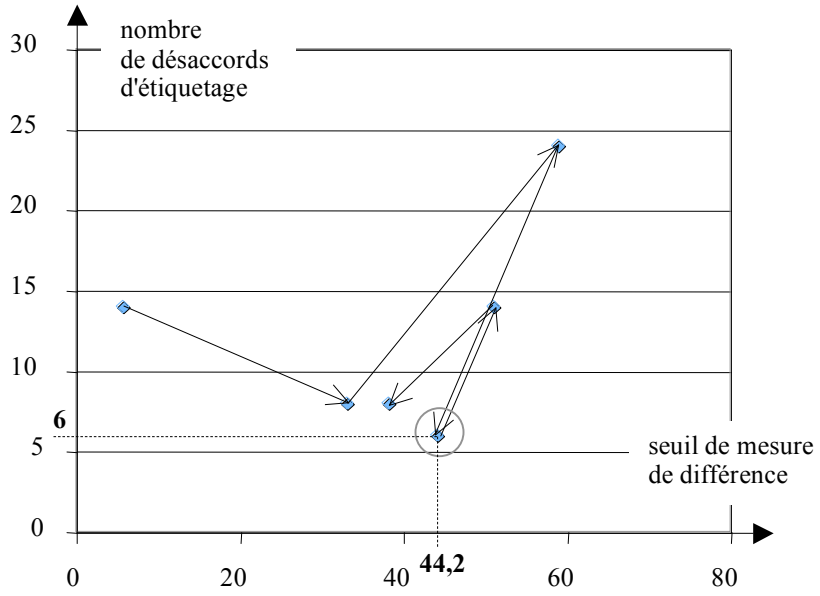
(les 2 mots les plus différents seraient un hapax de longueur `l_max` et un mot de longueur 1 et d'effectif `f_max`)

Pour approcher un minimum de la fonction à partir de faibles valeurs du seuil, le seuil initial est choisi légèrement supérieur à la borne inférieure (1). L'algorithme est classique, et procède par dichotomie, en rapprochant les bornes haute et basse tant que leur différence est supérieure à 1 et que la valeur de la fonction est supérieure à 0, avec une limite de 5 cycles.

Voici un exemple d'exécution sur un segment extrait du Monde du 15 avril 2004 :

M. Bush évoque une nouvelle résolution de l'ONU qui permettrait à d'autres pays de se joindre à ceux qui participent à une coalition aujourd'hui de plus en plus déstabilisée

La borne haute est : racine carrée (3673 * 12) / 2 = 104,9
(3673 = effectif de "*de*", 12 = longueur de "*déstabilisée*")
Les flèches indiquent la chronologie de la recherche du minimum de la fonction :



Dans ce cas, un nombre minimal de désaccords d'étiquetage, égal à 6, a été obtenu pour un seuil de mesure de différence égal à 44,2.
Voici le résultat de l'étiquetage (les mots informatifs sont en gras) :

M. Bush évoque** une **nouvelle résolution** de l'**ONU** qui **permettrait** à d'**autres pays** de se **joindre** à ceux qui **participent** à une **coalition aujourd'hui** de plus en plus **déstabilisée

La complexité pratique de cet algorithme d'étiquetage est linéaire en temps selon le nombre de mots du document, car le nombre de cycles de la répétition «tant que» de la recherche de minimum de fonction est borné (on ne recherche pas la meilleure solution en un nombre inconnu de cycles, mais une bonne solution en un nombre de cycles donné).

2.2. Génération de candidats termes de structure contrôlée

La structure d'un candidat terme est contrôlée par des motifs fondés sur l'étiquetage mots **I**nformatifs (`I`) ou **n**on (`n`).

En utilisant la syntaxe des expressions régulières, on génère d'abord les candidats simples, de motif `I+` , en ne prenant pas les candidats simples hapax (ce

sont des mots, dont on connaît l'effectif), pour ne garder que les candidats simples répétés. Par exemple, à partir de ***nouvelle résolution***, de motif `I+`, on génère la combinatoire des termes possibles : ***nouvelle***, ***résolution***, ***nouvelle résolution***.

On génère ensuite les candidats doubles, de motif `I+n+I+`, à partir des candidats simples répétés, car les candidats doubles répétés sont composés de candidats simples répétés. Les candidats doubles hapax sont alors supprimés. Par exemple, à partir de ***nouvelle résolution*** *de l'****ONU***, de motif `I+n+I+`, on génère la combinatoire des termes possibles : ***résolution*** *de l'****ONU***, ***nouvelle résolution*** *de l'****ONU***.

On génère ensuite les candidats triples, de motif `I+n+I+n+I+`, à partir des candidats doubles répétés. Les candidats triples hapax sont alors supprimés. Par exemple, à partir de ***or*** *de la* ***Banque*** *de* ***France***, de motif `I+n+I+n+I+`, on génère la combinatoire des termes possibles : ***or*** *de la* ***Banque***, ***Banque*** *de* ***France***, ***or*** *de la* ***Banque*** *de* ***France***.

On continue ainsi tant qu'on trouve des candidats non hapax, comme dans les algorithmes gloutons.

Cet algorithme étant combinatoire, sa complexité pratique est polynomiale de degré supérieur à 1, mais cette combinatoire reste faible, car les suites de mots informatifs contigus font 1 à 3 mots, et les suites longues et répétées sont rares.

2.3. Élagage de l'ensemble des termes

L'élagage a été déjà réalisé pour les candidats hapax au fur et à mesure du déroulement de l'algorithme de génération des candidats.

Il reste à supprimer les termes inclus dans des termes de même effectif, car ils sont moins informatifs. Par exemple, si ***résolution*** et ***résolution*** *de l'****ONU*** ont chacun 3 occurrences, cela implique que ***résolution*** ne se trouve que dans ***résolution*** *de l'****ONU*** et est moins informatif que le terme double qui le contient.

2.4. Pondération des termes

Une indexation doit s'accompagner d'une pondération, qui servira dans le calcul du «ranking» des réponses à une requête. Nous proposons que cette pondération soit fondée sur une estimation de la place occupée dans le rendu du document, soit pour chaque terme : effectif * longueur.

Dans un document html dont on a calculé la structure (voir note 2), nous faisons aussi intervenir la position du paragraphe par rapport au corps de texte, avec un coefficient défini comme étant égal à 1 pour le corps de texte, à des nombres supérieurs à 1 pour les paragraphes singletons dans leur classe de MFM, et situés avant le corps. Dans ce cas, le poids d'un terme devient : $\sum$ effectif * longueur * coefficient, en sommant sur chaque occurrence du terme.

3. Résultats

Dans son édition du 15 avril 2004, les 94 documents de profondeur 0 et 1 du site du journal Le Monde (on est ici dans le cadre d'une indexation de site) contenaient 26 occurrences de la graphie «or» (en minuscules), toutes référant au métal jaune, ce qui nous permet de montrer la robustesse de l'algorithme sur les mots informatifs courts et sur l'intérêt d'éviter une stop-list. Par exemple :

```
       f   l     (f = effectif, l = longueur)
 0     8   5   I Bercy
 1     8   7   I cherche
 2  1353   1   n à
 3     4   8   I utiliser
 4  1500   2   n l'
 5    26   2   I or
 6  3673   2   n de
 7  2000   2   n la
 8    19   6   I Banque
 9  3673   2   n de
10   120   6   I France
```

Sur 26 occurrences de «or», 21 ont été étiquetées informatives, 3 non informatives, 2 sont restées indéterminées (pour le seuil du minimum de la fonction, les étiquettes selon les différences avec les mots précédent et suivant étaient divergentes).
Les termes obtenus à partir de la graphie «or» sont :

```
f*l  f  l     (f = effectif, l = longueur)        motif
275 11 25  or de la Banque de France              InnInI
 68  4 17  rentabiliser l'or                       InI
 44  4 11  tonnes d'or                            InI
 42 21  2  or                                     I
 30  2 15  500 tonnes d'or                       IInI
 26  2 13  utiliser l'or                          InI
 16  2  8  or c'est                               III
  8  2  4  L'or                                  II
```

Voici un exemple tiré de l'International Herald Tribune du 15 octobre 2004 (102 documents de profondeur 0 et 1) sur la graphie «war», mot informatif court en anglais :

```
 0     1   8   I SARAJEVO
 1    97   2   n In
 2  1875   3   n the
 3    48   5   I years
 4    26   6   I before
 5  1875   3   n the
 6     5   7   I Bosnian
 7    33   3   I war
 8  1013   2   n of
 9  1875   3   n the
10     7   5   I early
11     7   5   I 1990s
```

Sur 33 occurrences de «war», 21 ont été étiquetées informatives, 10 non informatives, 2 indéterminées.
Voici les termes obtenus à partir de la graphie «war» :

```
f*l  f  l                                 motif
 63 21  3 war                             I
 46  2 23 effective war on terror          IInI
 22  2 11 war in Iraq                      InI
 18  2  9 civil war                       II
```

En allemand, la graphie «war» est une forme de l'auxiliaire «sein» (être), et devait être étiquetée non informative. Voici un exemple tiré du Spiegel du 15 avril 2004 (130 documents de profondeur 0 et 1) :

```
 0   144   3   n Der
 1     2  13   I Softwareriese
 2    83   3   n war
 3     4   7   I zuletzt
 4    12   5   I wegen
 5  1213   3   n der
 6    50   5   n immer
 7     1   8   I häufiger
 8     1  12   I auftretenden
 9     1  17   I Sicherheitsmängel
10   603   2   n in
11  1055   3   n die
12    12   6   I Kritik
13     5   7   I geraten
```

Sur 83 occurrences de «war», 8 ont été étiquetées informatives, 72 non informatives, 3 indéterminées.

Voici un exemple en italien, tiré de La Stampa du 15 avril 2004, sur la graphie «Iraq» (107 documents de profondeur 0 et 1) :

```
 0     4   2   n Ci
 1     1   6   I stiamo
 2     1   8   I muovendo
 3   140   3   n con
 4     9   6   I quelle
 5     5   8   I autorità
 6   302   3   n che
 7    33   3   - all
 8     3   7   I interno
 9    54   4   n dell
10    11   4   I Iraq
11   525   1   n e
12    33   3   - all
13     3   7   I interno
14    70   5   n delle
15     8  10   I principali
```

```
16     3   8   I comunità
17     1   9   I religiose
18    54   4   n dell
19    11   4   I Iraq
20     1   9   I riteniamo
21     1   7   I abbiano
22     1  13   I autorevolezza
23   525   1   n e
24     2   8   I capacità
25   852   2   n di
26     1   7   I indurre
```

Sur 11 occurrences de «Iraq», 10 ont été étiquetées informatives, aucune non informative, 1 indéterminée.

4. Conclusion et perspectives

Nous avons proposé une méthode d'indexation de documents de l'internet, fondée sur l'extraction de termes répétés et de structure contrôlée. Le contrôle de structure se base sur un étiquetage des mots à 2 étiquettes : mot informatif (pouvant être un index) et mot non informatif (ne pouvant pas être index seul). Cet étiquetage n'utilise que des propriétés de longueur et d'effectif des mots, et est donc possible sans connaître la langue du texte. Nous pensons qu'une telle méthode est applicable dans le cadre de l'indexation des moteurs de recherche sur l'internet, par sa légèreté calculatoire, son indépendance des langues, par l'obtention d'index moins volumineux qu'en indexation full-text, et par l'amélioration de la précision des réponses, mais ce dernier point reste à valider.

La question de l'évaluation d'une méthode d'indexation reste ouverte : peut-on évaluer seul un maillon de la chaîne des traitements d'un moteur de recherche, ou faut-il plutôt évaluer la chaîne entière : crawling + indexation + traitement des requêtes + ranking des réponses ? Cette évaluation devra se faire en collaboration avec des utilisateurs.

D'autre part, nous explorons actuellement la voie d'étiqueter non pas les mots, mais les différences entre mots, et d'en déduire un étiquetage des mots.

Au sujet des langues alphabétiques plus ou moins agglutinantes (par exemple : finnois, mais aussi allemand, espagnol dans une certaine mesure), des mots ne sont plus délimités par des espaces; mais la méthode est transposable en opérant un découpage en morphèmes avant étiquetage (travaux en cours). Au sujet des langues non alphabétiques (par exemple : chinois, japonais, coréen), la méthode est aussi transposable (travaux en cours), le grain atome étant non plus le mot, mais le groupe de caractères répété (cf. [He et al. 02]). Dans l'optique d'indexer toutes les écritures existantes, il nous faut ainsi abandonner le concept de «mot», trop centré sur les écritures des langues européennes, au profit d'une «molécule physique de texte» définie pour toutes les écritures, molécule à segmenter pour obtenir des «atomes», atomes dont les séquences répétées et de structure contrôlée sont à indexer.

Enfin, il faut mettre au point un calcul des termes concernés par une requête donnée, par exemple, les termes incluant tout ou partie de la requête, l'adéquation requête - termes devant participer au ranking des réponses.

5. Références bibliographiques

[Ahonen 99] Ahonen-Myka Helena. Discovery of frequent word sequences in text. *The ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, Imperial College, London, 2002.
www.cs.helsinki.fi/u/hahonen/ahonenmyka_patws02.ps

[Bourigault 02] Bourigault Didier. Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. *Actes de la 9ème conférence annuelle sur le Traitement Automatique des Langues (TALN 2002)*, Nancy, 2002, 75-84.
www.univ-tlse2.fr/erss/textes/pagespersos/ bourigault/TALN02-Bourigault.doc

[Brin & Page 98] Brin Sergey & Page Lawrence, "The Anatomy of a Large-Scale Hypertextual Web Search Engine",Computer Networks and ISDN Systems, Vol. 30, n° 1-7, pp. 107-117, 1998.
citeseer.ist.psu.edu/brin98anatomy.html

[He et al. 02] Hongzhao He, Jianfeng Gao, Pilian He, and Changning Huang, Finding the Better Indexing Units for Chinese Information Retrieval. *In: SIGHAN 2002*. Taipei, TaiWan.
acl.ldc.upenn.edu/W/W02/W02-1804.pdf

[Salem 87] Salem André. *Pratique des segments répétés*. Publications de l'INaLF, collection "St.Cloud", Klincksieck, Paris, 1987.

[Salton 83] Salton, G. and McGill, M.J. *Introduction to modern information retrieval*. New York: McGraw Hill, 1983.

[Salton 93] Salton Gerard and Allan James. Selective Text Utilization and Text Traversal. *In UK Conference on Hypertext*, 1993, 131-144.

[Saussure 22] Saussure F. de. *Cours de Linguistique Générale*. Payot, Paris, (éd. 1974), 1922.

[Tesnière 59] Tesnière Lucien. *Éléments de syntaxe structurale*. Klincksieck (Paris), 1959.

[Vergne 03] Vergne Jacques. Un outil d'extraction terminologique endogène et multilingue. *Actes de TALN 2003*, tome 2, 2003, 139-148.
www.info.unicaen.fr/~jvergne/TALN2003/ JVergne-TAL2003multV23.pdf

[Vergne 04] Vergne Jacques. Découverte locale des mots vides dans des corpus bruts de langues inconnues, sans aucune ressource. Actes des JADT 2004, volume 2, 2004, 1158-1164.

[Zipf 35] Zipf George Kingsley. *The Psychobiology of Language, an Introduction to Dynamic Philology*. Houghton Mifflin, Boston, 1935.

[Zipf 49] Zipf George Kingsley. *Human Behavior and the Principle of Least Effort*. Harper, New York, 1949.