# Chapitre VIII. Étude comparative des algorithmes de segmentation thématique pour les textes prophétiques[*].

## 1. Introduction

Avec l'augmentation de la taille des documents dans les collections de textes, la recherche et l'extraction de passages pertinents connaissent un essor particulier depuis une dizaine d'années. Les documents longs ou ceux possédant une structure complexe ou même les documents courts contenant plusieurs thèmes sont un défi pour les algorithmes qui ne distinguent pas quelle partie d'un document s'apparie avec une requête donnée. La segmentation thématique est une nouvelle technique pour l'amélioration de l'accès à l'information, elle peut être définie comme la tâche de subdivision d'un document en plusieurs paragraphes thématiquement cohérents. En recherche d'information par exemple, avoir des documents thématiquement segmentés peut résulter en la récupération des segments de texte courts et pertinents qui correspondent directement à la requête d'un utilisateur au lieu de longs documents examiné avec soin par l'utilisateur pour trouver l'objet de son intérêt. Avoir des documents thématiquement segmentés peut aussi aider dans la tâche de résumé automatique des textes puisque un meilleur résumé peut être obtenu de la fusion des différents segments constituant le document. La motivation principale de cette étude est d'étudier l'efficacité des algorithmes à base de cohésion lexicale comme moyen de segmentation des textes arabe.

Le reste de ce chapitre est organisé comme suit: la Section 2 présente les travaux antérieurs dans le domaine; la Section 3 présente une vue d'ensemble des approches implémentées; la Section 4 présente la méthodologie d'évaluation; les résultats d'évaluation sont rapportés dans la Section 5 et finalement la Section 6 est réservée pour la discussion des résultats.

## 2. Travaux antérieurs de segmentation thématique pour la langue arabe

Au temps où un nombre considérable de recherches a été consacré à l'étude cette technique pour les langues anglaise et française, peu l'ont étudié pour d'autres langues comme l'arabe. Les questions spécifiques au traitement de la langue arabe ont été soulevés dans [Hasnah, 1996], [El Shayeb et al., 2007] et [Touir et al., 2008]. Dans le travail de [Hasnah, 1996], l'algorithme TextTiling a été utilisé avec succès pour segmenter des textes de journaux arabes qui n'ont aucune marque de paragraphes dans un objectif de recherche d''information. Ce travail a fournit l'évidence que cet algorithme est applicable à la langue arabe. Le travail de [El Shayeb et al., 2007] est une analyse comparative des différents algorithmes de segmentation thématique sur des textes d'informations arabes. Ce travail a montré qu'une amélioration considérable dans les résultats peut être accomplie avec la combinaison de deux algorithmes de segmentation thématique. Il a aussi illustré que l'utilisation des conjonctions de filtrage pour la réduction de l'erreur dans les systèmes présentés peut considérablement raffiner les résultats. Dans [Touir et al., 2008], les auteurs introduisent une technique de segmentation thématique qui utilise des techniques linguistiques basées sur les articles connecteurs, en conservant la sémantique. L'algorithme est basé sur une liste de connecteurs actifs et passifs entre les différents segments

[*] Une grande partie de ce chapitre a été publiée dans les articles suivants :

1. F.Harrag, A. Hamdi-Cherif, et M. Benmohammed, Evaluation of Lexical Cohesion Algorithms for Arabic Topic Segmentation, *Revue d'information scientifique et technique (RIST)*, ISSN 1111-0015, Vol.18, N.1, pp. 103-116, 2010.
2. F.Harrag, A. Hamdi-Cherif, et A. S. Al-Salamn, Comparative Study of Topic Segmentation Algorithms Based on Lexical Cohesion: Experimental Results on Arabic Language, *The Arabian Journal for Science and Engineering*, ISSN 1319-8025, Vol.35, N.2C, 183-202, December 2010.

du texte. Leur expérimentation consiste a comparé les résultats obtenus par l'algorithme avec celles des experts linguistiques. L'inconvénient majeur de cette étude empirique est le manque d'adaptation pour la formalisation et la généralisation. Le manque de recherche dans ce domaine nous a donc poussés à adopter les deux algorithmes de segmentation thématique *TextTiling* et *C99* pour une telle langue.

## 3. Approches implémentés

Dans cette section, deux algorithmes de segmentation thématique des textes sont décrits: *TextTiling* [Hearst, 1997] et *C99* [Choi, 2000]. Les deux systèmes sont basés sur la cohésion lexicale. L'algorithme *TextTiling* utilise la mesure de similarité *Cosine* entre les vecteurs des termes pour mesurer la densité de la cohésion entre blocs adjacents. L'algorithme C99 utilise aussi la mesure de similarité *Cosine* pour déterminer des ressemblances parmi les phrases du texte puis il projette ceux-ci graphiquement. Il applique alors des techniques de traitement d'image pour déterminer des frontières thématiques.

### 3.1 Pré-traitement des textes

L'étape de pré-traitement traite les flux d'entrée en enlevant les étiquettes et les ponctuations et en transformant les termes en lemmes. En premier lieu, nous allons construire des blocs de texte appelés « séquences lexicales ». Le texte de l'entrée est simplement une séquence de caractères avant le pré-traitement. C'est la responsabilité du pré-processor de transformer cette séquence en unités sémantiques dans la phase d'analyse lexicale. Ces unités peuvent être des mots simples tels que les mots programme et création, ou des expressions composées telles que Les États-Unis (par opposition à États et Unis). Comme il est montré dans Figure 8.1, les deux méthodes *TextTiling* et *C99* extraient les noms, les verbes et les adjectifs à partir des textes d'entrée. La deuxième étape de la phase de pré-traitement, les mots fonctionnels sont supprimés. Les mots fonctionnels (ou mots vides) sont des mots non utiles pour le système de segmentation comme les préfixes, les pronoms, les prépositions,…, etc. La troisième étape est l'application d'un programme de lemmatisation qui consiste à la transformation de tous les mots dérivés à leur racine singulière commune ou leur forme canonique. Ce processus est très utile pour réduire et compresser la structure d'indexation, il permet aussi de profiter des rapports sémantique/conceptuel entre les différentes formes de la même racine. Le stockage des mots significatifs qui seront utilisés dans la prochaine phase de calcul de similarité met fin à la phase de pré-traitement.

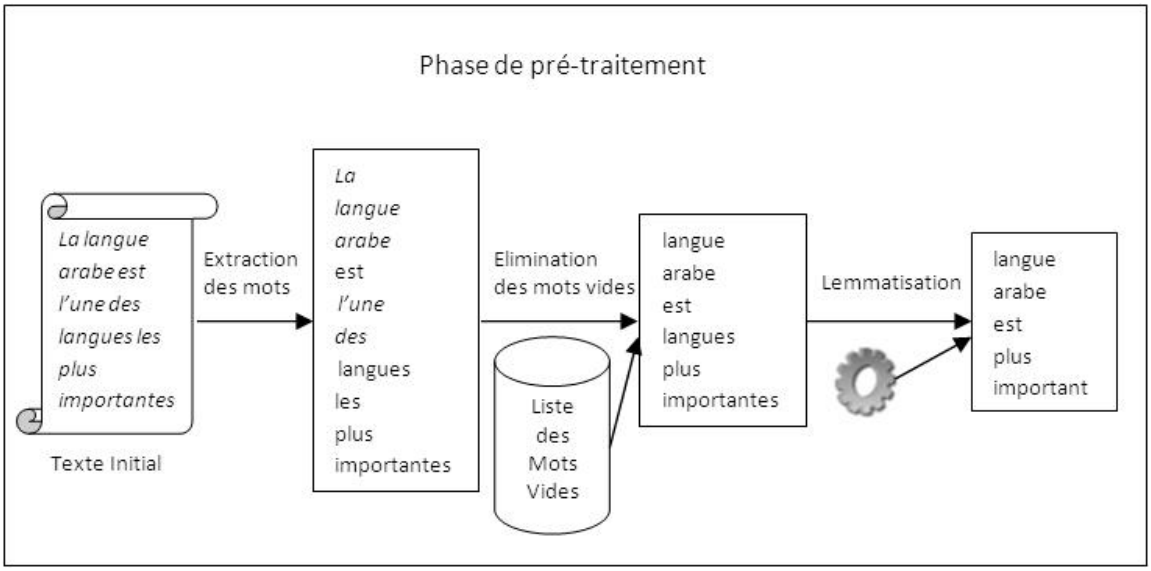


**Fig. 8.1.** La phase de pré-traitement.

### 3.2 L'Algorithme TextTiling

TextTiling est l'une des méthodes de segmentation thématique présenté par [Hearst, 1994]. L'idée de base de cette méthode est la recherche des partie de texte où le vocabulaire change d'un sous thème à l'autre. TextTiling réalise le découpage d'un texte en unité cohérente qui reflète sa structure thématique. Cet algorithme qui utilise la comparaison des paires de blocs adjacents d'un texte, fait l'hypothèse que deux blocs sont similaires s'ils traitent le même sujet. La similarité étant évaluée par un calcul de cosinus sur les blocs, elle est déterminée par un modèle de l'espace vectoriel [Amini, 2001].

#### 3.2.1. Détail de la méthode TextTiling

La Figure 8.2 ci-dessous montre que la première étape de la méthode TextTiling consiste au découpage du document en blocs fixes de phrases (généralement de 3 à 5 phrases). Dans la deuxième étape, les blocs adjacents sont comparés deux à deux et une valeur de similarité est attribuée à chaque paire de blocs. Dans la troisième étape, la suite résultante des valeurs de similarité est mise sous une forme d'un graphe de similarité et les points de ce graphe sont examinés pour la détermination des pics et des vallées. La quatrième étape consiste à la mise en place des passages en se basant sur les ruptures des thèmes. Les valeurs de similarité élevées impliquent que les blocs adjacents se suivent de façon logique (c.-à-d. : ayant le même thème), ces valeurs sont représentées sous forme de pics. Les valeurs de similarité faibles indiquent une potentielle limite entre les blocs et donc sont représentées sous forme de vallées [Khalis, 2006]. Un pic correspond donc à deux blocs fortement liés thématiquement, alors qu'une vallée correspond à une rupture de thème (considérée comme une limite entre deux blocs thématiquement différents).

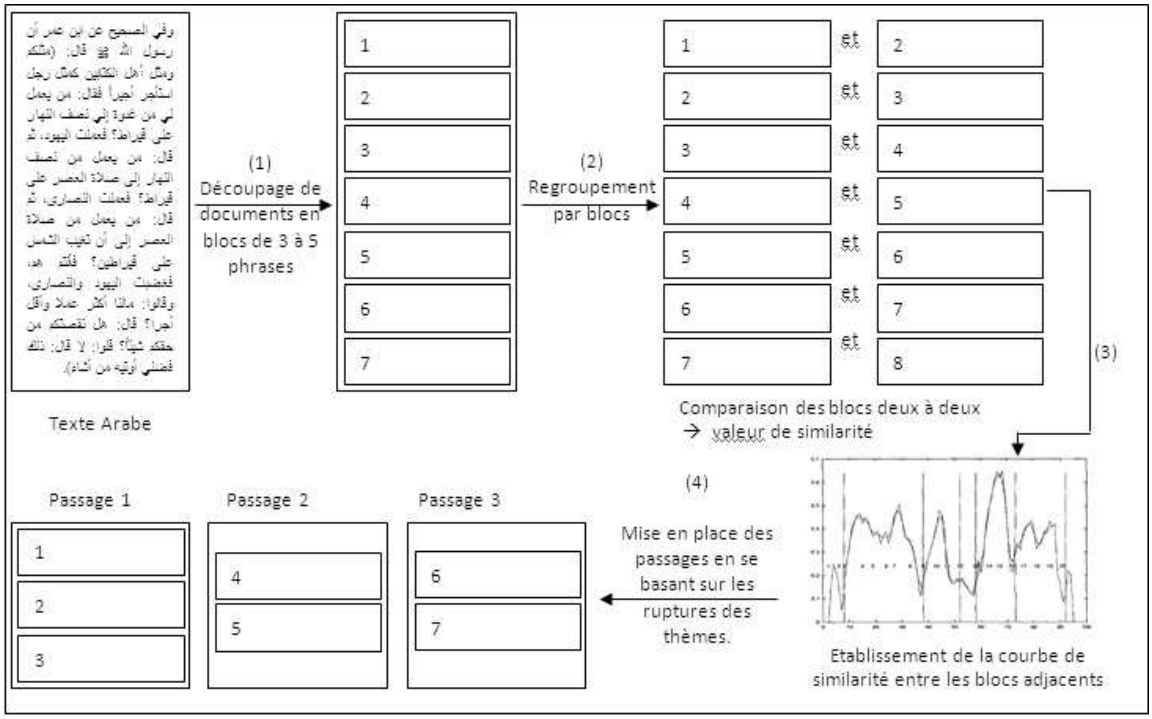


**Fig. 8.2.** Le principe de la méthode TextTiling.

#### a. Découpage physique

La longueur de phrases peut varier. Pour cela le texte issu de pré-traitement (contient les lemmes) est divisé en petites unités de taille fixe (par exemple : 20 mots) appelés *pseudo phrases*, eux même regroupés en blocs de taille fixée par l'utilisateur. La structure des

paragraphes réels ainsi que des phrases n'est pas pris en considération car leur longueur peut être fortement irrégulière conduisant à des comparaisons déséquilibrées.

### b. Calcul de similarité

La comparaison entre blocs est faite par un calcul de similarité entre chaque paire de blocs selon l'équation Cosine définie par:

$$Cosine\ (b_1, b_2) = \frac{\sum_{w \in b_1 \cap b_2} TFIDF_{w,b_1} \cdot TFIDF_{w,b_2}}{\sqrt{(\sum_{w \in b_1} TFIDF_{w,b_1}{}^2) \cdot (\sum_{w \in b_2} TFIDF_{w,b_2}{}^2)}} \qquad (8.1)$$

Tel que: *w* est un terme, $b_1$ et $b_2$ sont deux blocs adjacents, $TFIDF_{w,b_1}$ est le poids de *w* dans $b_1$ et $TFIDF_{w,b_2}$ est le poids de *w* dans $b_2$. La formulation du poids TFIDF tient compte du nombre d'apparitions du terme dans le bloc et du nombre de bloc qui contient ce terme dans le document (elle est appelée TFIDF *Term Frequency, Inverse Document Frequency*). Cette formulation fait l'hypothèse qu'un terme est important pour un bloc donné, s'il apparaît souvent dans ce bloc et que peu de bloc le contient [Salton & Buckley, 1988]. Cette pondération est définie de la manière suivante:

$$TFIDF(w, b) = TF_{w,b} \cdot IDF_{w,b} = TF_{w,b} \cdot \left( \left( \log_2 \frac{N}{DF_w} \right) + 1 \right) \qquad (8.2)$$

Tel que: *w* est un terme, *b* est un bloc, $TF_{w,b}$ est le nombre d'apparitions de *w* dans *b*, $DF_w$ est le nombre de bloc qui contiennent *w* et *N* est le nombre total de bloc dans le document. Le graphe de similarité est établi après le calcul des valeurs de similarité de toutes les paires de blocs.

### c. Calcul de score de profondeur (Depth Score)

A partir des scores de similarité, un score de cohésion ou de profondeur est assigné à chaque creux (ou gap) en fonction des scores de similarité des creux voisins comme il est montré dans la Figure 8.3. Le score de cohésion représente donc la somme des différences entre le sommet du pic et les creux des vallées voisines [Hearst, 1997]. L'algorithme de calcul du score de cohésion est donné ci-dessous:

- On commence par le premier creux entre deux blocs.
- On mémorise le score de similarité associé au *g(i), g(i-1), g(i+1)* ;tel que *g(i)* est le creux du point *i*.
- On vérifie les scores de similarité des creux voisins [gauche de *g(i-1)* et droite de *g(i+1)*].
- Si c'est plus haut, on continue et on examine le score du creux voisin jusqu'à ce que le score soit plus bas.
- On soustrait le score de similarité du creux initial du score maximum des similarités rencontrées.
- *depth (g(i)) = [g(i-1)-g(i)]+ [g(i+1)-g(i)].*
- répéter pour tous les gaps.

**Algorithme 8.1.** Algorithme de calcul du score de cohésion.

Les scores de cohésion ne sont calculés que pour les creux qui sont des minimax locaux pour la fonction de similarité.

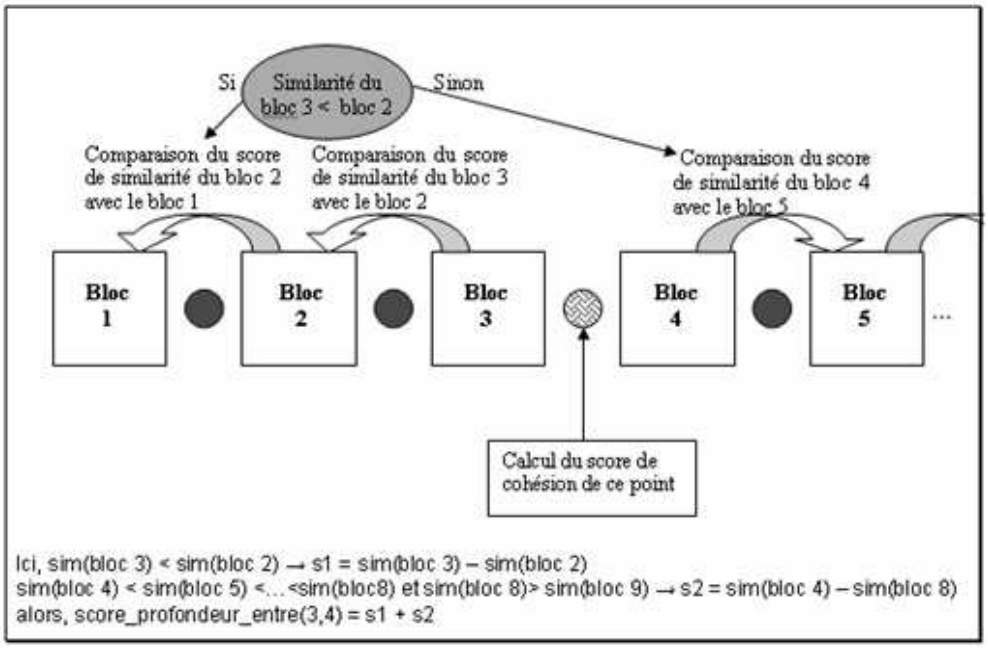


**Fig. 8.3.** Calcul du score de cohésion (profondeur).

La Figure 8.4 ci-dessous, montre trois cas critiques pour le calcul du score de cohésion. Dans le cas (a) le score est calculé selon la formule: *Depth(a2)= (Ya1-Ya2) + (Ya3-Ya2)*, c'est un cas normale. Deux problèmes sont rencontrés lors de calcul du score de profondeur, ils sont montrés dans les cas b et c. Dans le cas (b): les petites vallées vont créer une perturbation dans la détermination des ruptures. Dans le cas (c): les deux vallées C2, C3 forme un *plateau* d'où la difficulté d'identifier les ruptures.

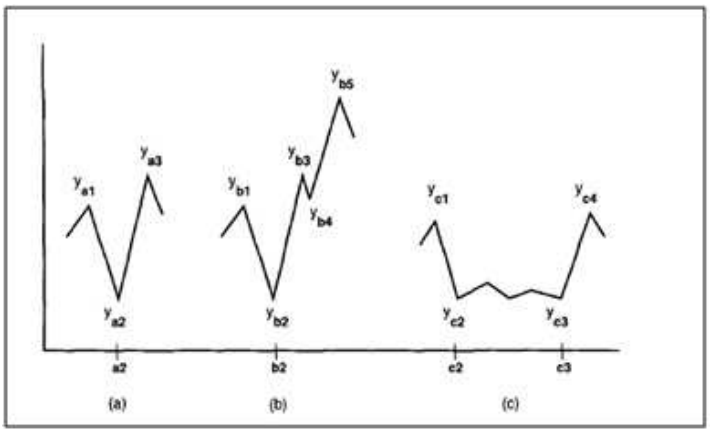


**Fig. 8.4.** Le score de cohésion dans 3 situations critiques

### d. Convolution

Pour une implémentation pratique, les algorithmes ont besoin d'être améliorés. On a besoin d'une fonction de convolution à appliquer sur le score de cohésion pour répondre aux difficultés discutées dans la section précédente. L'algorithme d'application de convolution est donné ci-dessous:

- Pour chaque creux ***g(i)*** de score ***S :***
  - Mémoriser le score $S_1$ pour le creux ***g(i-1).***
  - Diviser $S_1$ sur 2: $S_1 \leftarrow S_1/2$.

- Mémoriser le score $S_2$ pour le creux $g(i+1)$.
- Diviser $S_2$ sur 2: $S_2 \leftarrow S_2/2$.
- Prendre la moyenne de ces scores : $\bar{S} = (S_1 + S_2 + S_3)/3$

▪ Répéter cette procédure n fois (n est le nombre de creux).

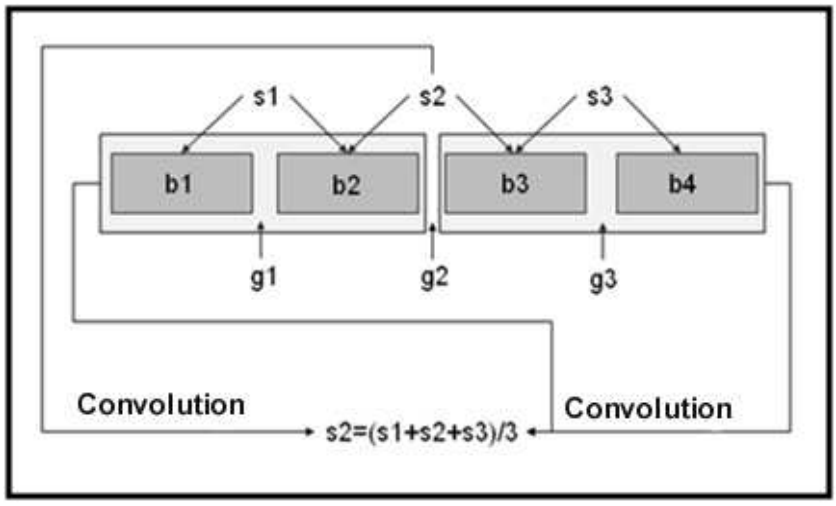


**Fig. 8.5.** Application de la convolution.

### e. Sélection des frontières (limites des segments)

Les meilleurs points pour la sélection des limites de segmentation sont basés sur le score de cohésion et la sélection des creux. On estime la moyenne $\bar{S}$ et la déviation standard $\delta$ du score de cohésion et on prend tous les creux qui ont un score de cohésion supérieur à $\bar{S} - \delta/2$ [Hearst, 1997] ; tel que :

$$\delta = \frac{(DepthScore\ [i] - \bar{S})^2}{2} \qquad (8.3)$$

## 3.3 L'algorithme C99

L'algorithme C99 proposé par [Choi, 2000], utilise une mesure de similarité entre chaque unité textuelle. L'idée de base de cette méthode est que les mesures de similarité entre des segments de textes courts sont statistiquement insignifiantes, et que donc seul des classements locaux (voir ci-dessous) sont à considérer pour ensuite appliquer un algorithme de catégorisation sur la matrice de similarité. L'algorithme C99 utilise la mesure de similarité *Cosine* pour déterminer des ressemblances parmi les phrases du texte puis il projette ceux-ci graphiquement. Il applique des techniques de traitement d'image pour déterminer des frontières thématiques.

### 3.3.1. Détail de la méthode C99

La description générale de la méthode C99 est illustrée dans la Figure 8.6. Dans un premier temps, une matrice de similarité est construite, représentant la similarité entre toutes les phrases du texte à l'aide de la mesure de similarité Cosinus, calculée pour chaque paire de phrases du texte, en utilisant chaque mot commun entre les phrases, et après « nettoyage » du texte : suppression des mots vides et lemmatisation. On effectue ensuite un « classement local », en déterminant pour chaque paire d'unités textuelles, le rang de sa mesure de similarité par rapport à ses $m \times n - 1$ voisins, $m \times n$ étant le masque de classement choisi. Le rang est le nombre d'éléments voisins ayant une mesure de similarité plus faible, conservé sous la forme d'un ratio $r$ afin de prendre en compte les effets de bord. Enfin, la dernière étape détermine les limites de

chaque segment de la même manière que l'algorithme *Dotplotting* [Reynar, 2000] emploie la maximisation. En effet on cherche à déterminer quelle configuration offre la plus grande densité, en recherchant une nouvelle limite thématique à chaque étape. Les segments sont alors représentés par des carrés le long de la diagonale de la matrice de similarité modifiée avec les classements locaux. Pour chaque segment de la répartition proposée à une étape de la segmentation on considère son aire notée $a_k$ et son poids $s_k$ qui est la somme des tous les rangs des phrases qu'il contient. On calcule alors la densité $D$ de la configuration. L'algorithme s'arrête lorsque la densité de la meilleure répartition proposée est suffisamment faible, ou lorsque le nombre de frontières thématiques déjà déterminé est atteint.

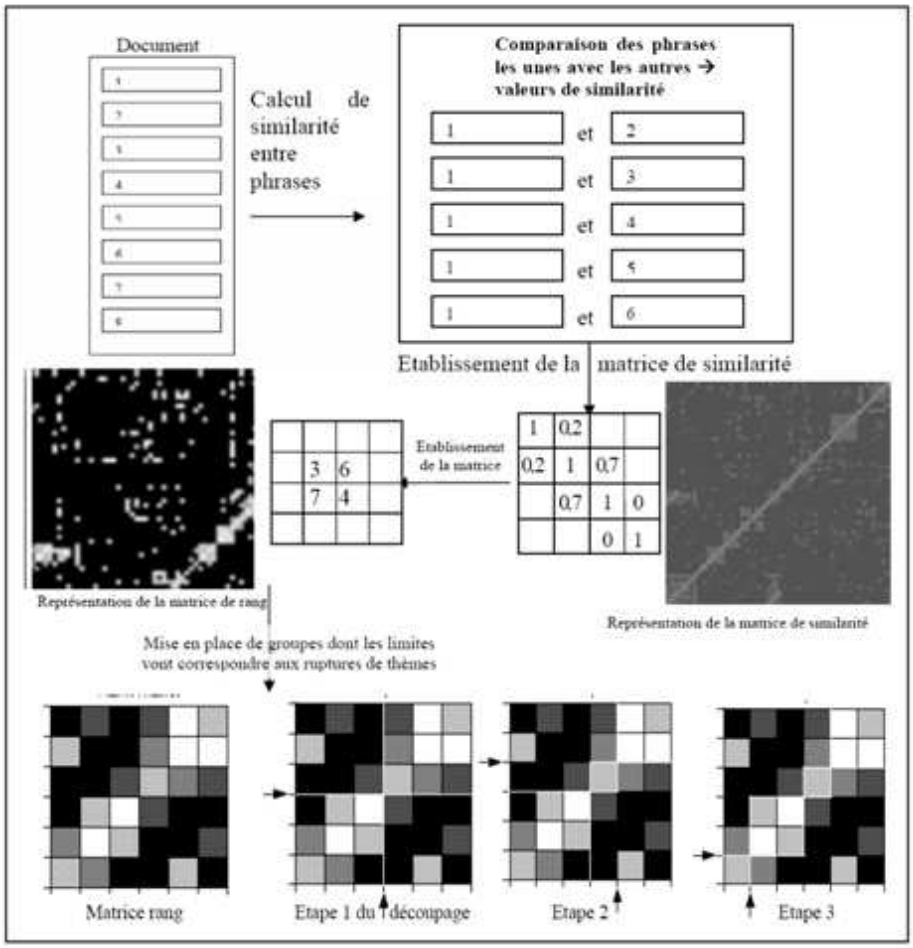


**Fig. 8.6.** Principe générale de la méthode C99.

### a. Le découpage physique

Pour effectuer la segmentation seuls les mots compris dans des catégories ouvertes c'est-à-dire les substantifs, les verbes et les adjectifs seront pris en compte. Ensuite, les mots restants sont lemmatisés en utilisant un programme qui retrouve la forme non fléchie conventionnelle de chaque mot [Khalis, 2006]. L'algorithme C99 prend une liste de phrases composées des lemmes issus du prétraitement. Un dictionnaire de fréquences des lemmes est construit pour de chaque phrase, il est représenté sous forme d'un vecteur que ses cordonnées sont les valeurs de fréquences des lemmes.

### b. Matrice de similarité

La similarité entre toutes les paires de phrases *x*, *y* est calculée en utilisant la mesure du Cosinus, cette mesure est appliquée pour toutes les paires de phrases afin de générer la matrice de similarité.

$$Sim\ (x, y) = \frac{\sum_j f_{x,j} \cdot f_{y,j}}{\sqrt{\sum_j f_{x,j}{}^2 \cdot \sum_j f_{y,j}{}^2}} \qquad (8.3)$$

Tel que: $f_{x,j}$ est la fréquence du mot *j* dans la phrase *x* et $f_{y,j}$ est la fréquence du mot *j* dans la phrase *y.*

La matrice de similarité est symétrique et les éléments de la diagonale sont tous égaux a 1, en effet la similarité entre la phrase *x* et la phrase *y* est la même qu'entre la phrase *y* et la phrase *x*. De plus, la similarité entre la phrase *x* et la phrase *x* est égale 1 puisque se sont les mêmes comme il est montré dans la Figure 8.7.

| | | | |
|---|---|---|---|
| 1 | 0.2 | | |
| 0.2 | 1 | 0.7 | |
| | 0.7 | 1 | 0 |
| | | 0 | 1 |

**Fig. 8.7.** Exemple d'une matrice de similarité.

A partir de cette matrice, une visualisation graphique est possible en représentant les valeurs de similarités importantes par des pixels de couleur clairs comme il est montré dans Figure 8.8.

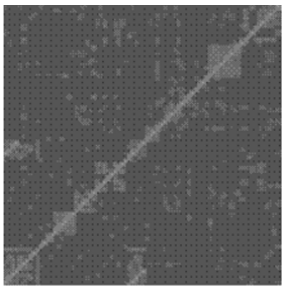

**Fig. 8.8.** Représentation graphique de la matrice de similarité.

Les pixels de bas gauche et de haut droite représentent la similarité réflexive des première et dernière phrases respectivement. La matrice contient des régions de forme carrée claires le long de la diagonale. Ces régions représentent les segments du texte qui ont la même thématique.

### c. Matrice de Rang

Les données fournies par la matrice de similarité peuvent être discutés car le coefficient de cosinus est peu fiable. En effet, pour de petits textes, la valeur absolue de la mesure de similarité est peu fiable, une occurrence additionnelle d'un mot commun (reflétée dans le numérateur) provoque des disproportions accrues dans la mesure de similarité à moins que le dénominateur (relatif à la longueur du segment) soit grand. Ainsi, dans le contexte de segmentation des textes où un segment a typiquement moins de 100 unités informatives, on peut employer la métrique Cosinus pour seulement estimer l'ordre de la similarité entre les phrases, par exemple la phrase *A* est plus similaire à la phrase *B* qu'a la phrase *C*. En outre, l'emploi de la langue change à travers le document, l'introduction par exemple, est moins cohésive qu'une section liée à un thème particulier. Donc, il est inadéquat de comparer les valeurs de similitude d'une région a l'autre.

Dans le cas d'une analyse statistique non paramétrique, on peut comparer le rang de l'ensemble des données quand le comportement qualitatif est similaire alors que les quantités absolues sont peu fiables. La mesure alternative du « Rang » est donc proposée, elle est basée sur la mesure du coefficient Cosinus. A partir de la matrice de similarité, un classement est effectue et une matrice de rang est établie. Pour cela chaque valeur de la matrice est remplacée par son rang dans le contexte local pour produire une seconde matrice: la matrice de rang. Le rang entre *x* et *y* est la proportion de paires de phrases voisines qui ont une valeur de similarité inferieure à la mesure *Cosinus(x, y)*. La Figure 8.9 montre un exemple de rangement d'une image en utilisant un masque de rang 3*3.

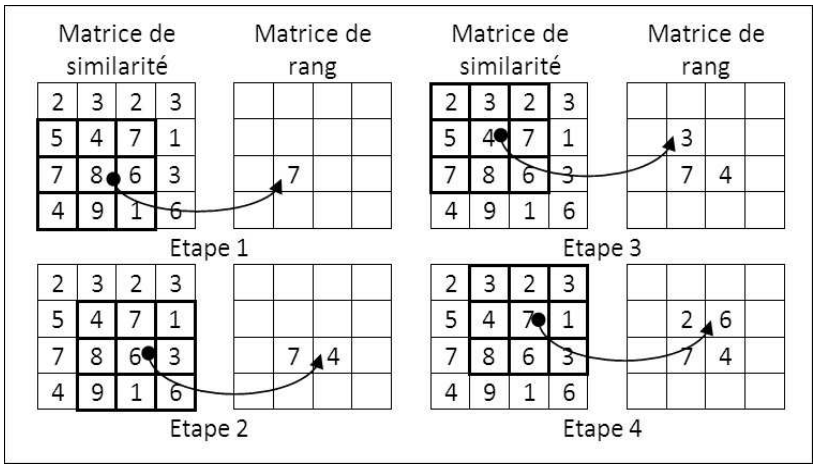


**Fig. 8.9.** Calcul de la matrice de rang à partir de la matrice de similarité.

Pour la segmentation, un masque de rang 11*11 est utilisé. Pour éviter des problèmes de normalisations (pour les similarités se trouvant sur les bords de la matrice), le rang est exprimé sous forme d'un ratio :

$$r = \frac{nombre\ d'élémentsavec\ une\ valeur\ inferieure}{nombre\ d'éléments\ comparés} \qquad (8.4)$$

Qui correspond au nombre d'éléments ayant une similarité inferieure à celle dont on calcule le rang divisé par le nombre d'éléments comparés (au maximum 11*11). Pour démontrer les conséquences de la mise en place d'une matrice de rang, la matrice de rang est représentée visuellement de façon analogue à la matrice de similarité, La Figure 8.10 montre que la différence entre les deux matrices est très nette, les contrastes de la matrice de rang sont plus accentués que celles de la matrice de similarité.

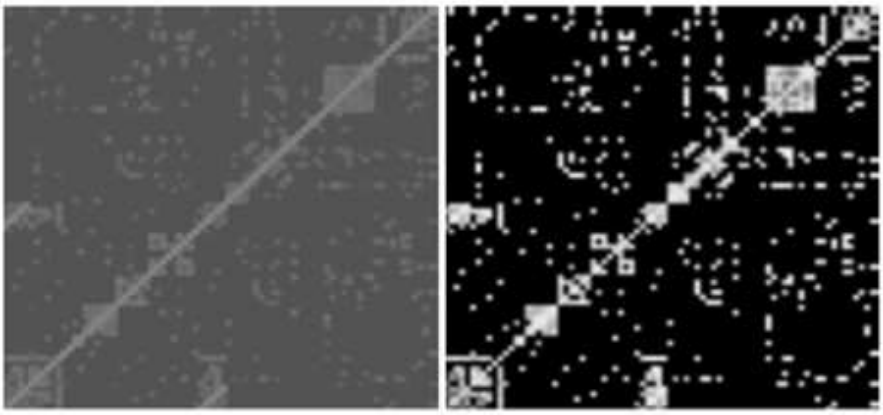

**Fig. 8.10.** Représentation visuelle : à gauche de la matrice de similarité ; à droite de la matrice de rang.

### d. Extraction des zones thématiques

Dans la dernière étape, l'algorithme de groupement des phrases va permettre de localiser les changements de thèmes. La méthode est basée sur l'algorithme de maximisation de Reynar [Reynar, 2000]. Un segment de texte est délimité par deux phrases *x* et *y,* il est donc représenté comme une région carrée le long de la diagonale de la matrice de rang et l'aire interne de ce segment est calculée de la façon suivante:

$$a_{xy} = (y - x + 1)^2 \qquad (8.5)$$

La densité interne des segments correspond à la somme des valeurs de la matrice de rang dans toutes les régions de forme carrées, c'est-a-dire les segments, divisés par la somme de leurs aires :

$$D = \frac{\sum_{k=1}^{m} s_k}{\sum_{k=1}^{m} a_k} \qquad (8.6)$$

Où $s_k$ est la somme des valeurs de la matrice de rang appartenant au segment $k$, $a_k$ est l'aire interne de ce segment et $k$ est le numéro du segment dans la liste $B$, tel que $B= \{B_1,\ldots,B_m\}$ représente la liste des m segments cohérents du texte. Initialement, le texte tout entier est placé dans $B$ comme un seul segment cohérent, puis à chaque étape un segment de $B$ est coupé en deux. Le point de coupure est potentiellement une localisation qui maximise la densité $D$. Le nombre $m$ de segments à générer est déterminé automatiquement. $D^{(n)}$ est la densité interne de $n$ segments et $\delta D^{(n)} = D^{(n)}-D^{(n-1)}$ est le gradient. Pour un document avec $b$ frontières possibles, $b$ étapes de regroupement génère $\{D^{(1)},\ldots,D^{(b+1)}\}$ et $\{\delta D^{(2)},\ldots, \delta D^{(b+1)}\}$. Une réduction exceptionnelle du gradient signifie que les regroupements maximisant la densité interne ont été trouvés .En pratique, le graphe du gradient est lissé en utilisant une convolution (masque {1, 2, 4, 8, 4, 2, 1}) en premier afin d'aplatir les pics locaux. Soit μ et ν qui correspondent respectivement à la moyenne et la variance du gradient $\delta D(n)$, $n \in \{2, \ldots, b + \&\}$. $m$ est obtenue par l'application du seuil $(u + c \times \sqrt{v})$ à $\delta D$.

## 4. Méthodologie d'évaluation

Dans cette section, nous donnons le détail des métriques d'évaluation déterminant les performances du système de segmentation ainsi que le corpus de textes arabes utilisé pour cette fin. L'efficacité des algorithmes de segmentation implémentés a été évaluée en utilisant la méthode de jugements des lecteurs et la méthode du rappel/précision développées dans [Hearst, 1997]. Ces méthodes mesurent la proportion de désaccords entre les segmentations d'un groupe de lecteurs et celles produites par les deux algorithmes.

### 4.1 Critères d'évaluation

L'évaluation de la segmentation thématique peut se faire de plusieurs manières :

- Par comparaison avec des jugements humains : aucun corpus segmenté de taille suffisante n'est cependant disponible jusqu'à ce jour; des propositions ont été faites pour la constitution d'un tel corpus et pour évaluer la qualité des jugements humains [Passonneau et Litman, 1993][ Carletta, 1996][Hearst, 1997].
- Par rapport à des marques déposées par l'auteur du texte (cette procédure n'est pas fiable car toute segmentation est subjective [Passonneau et Litman, 1993], la position des marques de segmentation dépend du point de vue du lecteur) ;
- Par rapport à des marques «certaines» à retrouver (limites entre documents d'un corpus par exemple);
- Par son impact sur une tâche particulière (évaluation fonctionnelle), la recherche d'informations par exemple.

En plus de la méthode de jugements des lecteurs et de la méthode de rappel/précision utilisées dans ce mémoire, d'autres métriques sont connues dans la littérature de la segmentation thématique. On peut citer la métrique de Beeferman [Beeferman et al. 1999] qui essaie d'adresser les inconvénients de la méthode rappel/précision. Cette méthode est basée sur une métrique d'évaluation probabiliste $P_k$. Cependant, dans une publication récente, Pevzner et Hearst [Pevzner et Hearst 2002] ont mis en valeur plusieurs défauts de la métrique $P_k$. Les auteurs montrent par plusieurs preuves et scénarios de segmentation qui leur métrique alternative appelée *WindowDiff*, allège ces problèmes et fournit des performances plus justes et plus exactes.

### 4.2 Corpus d'évaluation

Dans notre évaluation, nous adaptons des approches antérieures qui ont été étudiées pour la langue anglaise. Un effort a été fait pour les appliquer à la langue arabe. Par exemple, pour le travail de [Hearst, 1997] qui tend à tester l'algorithme TextTiling, l'évaluation a été basée sur des jugements obtenus de sept lecteurs pour chacun de 12 articles qui satisfaites un critère de longueur (entre 1,800 et 2,500 mots) et qui contient de petites démarcations structurelles. Les juges ont été simplement demandés à marquer les limites des paragraphes où on croit qu'il y a un changement de thème. Les juges n'ont aucune autre information explicite qui concerne le détail de la segmentation.

Les systèmes de segmentation analysés ont été évalués en utilisant un ensemble de cinq textes arabes. Nous avons basé notre évaluation sur la comparaison des résultats obtenus avec les jugements d'un groupe de sept lecteurs. Ces textes présentent une variété de thèmes (médical, littéraire, scientifique et islamique). La longueur moyenne des textes utilisée pour cette évaluation est entre 600 et 2000 mots. Chaque phrase est séparée de la suivante par un retour chariot et il n'y a aucune indication typographique qui signale la présence d'un paragraphe dans le texte original. Les lecteurs sont simplement invités à définir les paragraphes dans lesquels il ya un changement de thème. Cette opération reste subjective pour chaque lecteur.

Avant toute analyse, les textes du corpus d'évaluation ont été lemmatisés en utilisant le programme *Detect-Root* [Darwish, 2002] et un ensemble de mots vides a été supprimé de chaque texte. La Lemmatisation des textes du corpus est une procédure classique dans la segmentation automatique des textes parce qu'elle permet d'augmenter la proportion de mots identiques. Elle est particulièrement utile pour la langue arabe parce qu'elle permet d'obtenir la même racine pour différentes formes d'un verbe conjugué.

## 5. Résultats d'évaluation

Dans cette section, nous présentons les résultats d'évaluation de chaque algorithme sur un ensemble de textes arabes en utilisant les métriques d'évaluation susmentionnée. Nous déterminons l'efficacité de l'algorithme TextTiling en le comparant avec un autre algorithme de segmentation basée sur la cohésion lexicale, à savoir l'algorithme C99[3]. La première étape consiste à appliquer les routines de pré-traitement discutées dans la Section 3.1 sur notre corpus de textes arabes. Dans la deuxième étape, nous avons exécuté les deux algorithmes de segmentation décrits dans les Sections 3.2 et 3.3 en employant des valeurs par défaut pour les paramètres de chaque algorithme. Pour simplifier les analyses, nous fournissons aux algorithmes le nombre de segments qu'ils devraient identifier, les paramètres de chaque algorithme ont été fixés en fonction des indications donnée son auteur [Hearst, 1997][Choi, 2000].

[3] Nous utilisons notre implémentation des algorithmes TextTiling et C99 inspirée de l'implémentation en langage Java du Frederic Choi disponible pour téléchargement gratuit sur www.cs.man.ac.uk/~choif.

### 5.1. Méthode de Jugements des Lecteurs

Le schéma de la Figure 8.11 montre les limites faites par les sept lecteurs sur les textes. Ce schéma nous aide à illustrer les tendances générales des évaluations des lecteurs, et également à montrer où/et combien de fois ils sont en accord ou en désaccord. Par exemple, tous les lecteurs sauf le quatrième ont marqué une frontière au paragraphe 7. Ce lecteur en désaccord avec les autres a délimité la frontière au paragraphe 10. L'ensemble des frontières pour lesquelles les lecteurs sont tous en accord sont les suivants: {12, 20, 22, 31, 33, 37, 38, 50}. Par contre, il y a un désaccord pour les frontières suivantes: {1, 15, 18, 41,43, 44, 45 ...}.

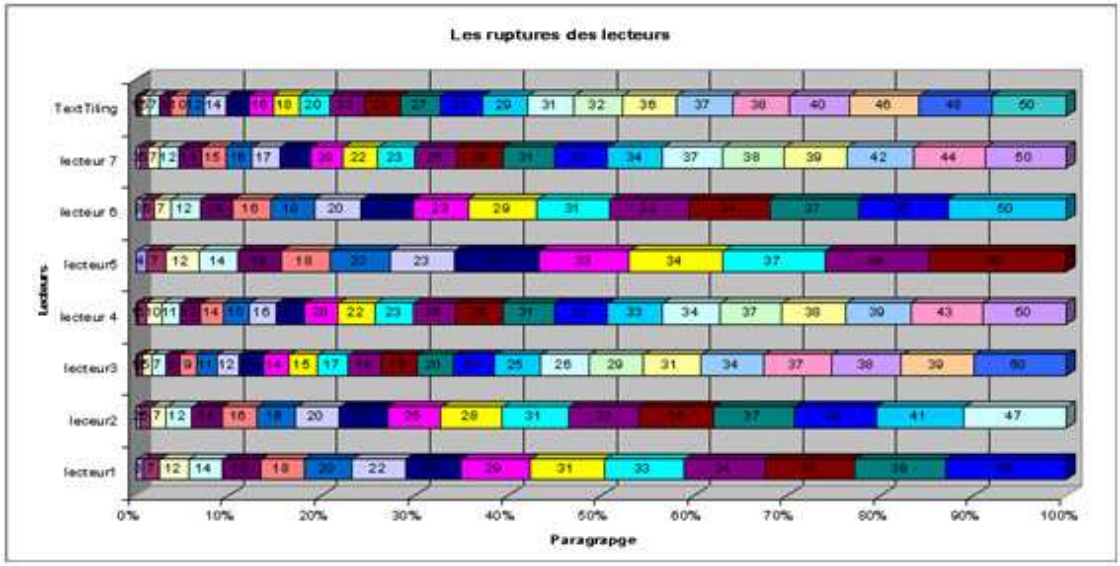


**Fig. 8.11.** Les ruptures des lecteurs contre les ruptures des algorithmes.

D'après [Passonneau et Litman, 1993], si quatre ou plus sur sept lecteurs marquent la même frontière, la segmentation s'avérée. Mais, deux années après [Litman et Passonneau, 1995], ils ont montré que trois lecteurs sont considérés suffisent pour classer u point comme une frontière "principale". [Isard et Carletta, 1995] et [Carletta, 1996] précisent l'importance de tenir en compte l'accord fortuit et prévu en calculant si les lecteurs convient de manière significative. A cette fin, Ils conseillent d'utiliser le coefficient de *Kappa* (*K*). D'après [Carletta, 1996], *K* mesure par paire, l'accord parmi un ensemble de lecteurs faisant des catégories de jugements, il est calculé selon l'équation suivante :

$$k = \frac{P(A)-P(E)}{1-P(E)} \qquad (7.7)$$

Où P(A) est la proportion de fois que les lecteurs conviennent et P(E) est la proportion de fois où on s'attendrait à ce qu'ils conviennent par hasard. Le coefficient peut être calculé en faisant par paires des comparaisons contre un expert ou en comparant à une décision de groupe. [Carletta, 1996] déclare également que si K est supérieur à 0.8, alors la segmentation est bonne, et si K est supérieur à 0.67 et inferieur à 0.8, alors cela permet de donner des conclusions expérimentales acceptables. Les coefficients trouvés par [Isard et Carletta, 1995] se sont étendus de 0.43 à 0.68 pour trois lecteurs, et ceux trouvées par [Carletta, 1996] sont étendus de 0.65 à 0.90 pour quatre lecteurs segmentant des phrases.

Dans notre évaluation, nous concéderons que trois jugements en accord sont acceptables pour considérer la frontière juste. A partir de la Figure 8.11, l'ensemble des frontières acceptables est le suivant: {1, 3, 5, 7, 12, 14, 15, 16, 18, 20, 22, 23, 29, 31, 33, 34, 37, 38, 50}. On peut calculer le coefficient Kappa comme il est montré dans le Tableau 8.1 ci-dessous. La comparaison de nos résultats avec celles obtenus par Hearst [Hearst, 1997] à partir de l'application de l'algorithme *TextTiling* sur un corpus anglais a montré que notre segmentation est acceptable.

| P(A) | P(E) | K | K(H) | Remarque |
|---|---|---|---|---|
| 0.7894 | 0.2106 | **0.7332** | 0.647 | **Acceptable** |

**Tableau 8.1.** Résultats de calcul du coefficient Kappa.

### 5.2. Méthode de Rappel / Précision

Dans l'expérience suivante, les deux mesures rappel et précision, classiquement utilisés dans la recherche d'information, détaillés dans [Baeza-Yates & Ribeiro-Neto, 1999], ont aussi été employés pour évaluer les algorithmes de segmentation. Dans le contexte de segmentation thématique, la précision est définie comme:

$$\text{Précision :} \quad P = \frac{Nombre\ de\ frontières\ correctement\ détectées\ par\ le\ système}{Nombre\ total\ de\ frontières\ générées\ par\ le\ système} \qquad (8.8)$$

Tandis que le rappel est défini comme:

$$\text{Rappel :} \quad R = \frac{Nombre\ de\ frontières\ correctement\ détectées\ par\ le\ système}{Nombre\ total\ de\ frontières\ de\ référence} \qquad (8.9)$$

Pour la tâche de segmentation thématique, les deux mesures (Rappel et Précision) combinées définissent ce qu'on appelle la Mesure F, cette mesure est utilisée pour donner la possibilité de comparer nos méthodes implémentées avec d'autres travaux dans le même domaine:

$$\text{Mesure } F_1 : \quad F_{1=}\frac{2 \cdot R \cdot P}{R+P} \qquad (8.10)$$

Les valeurs de Rappel et Précision pour les deux algorithmes nous donnent une idée générale sur l'échec de ces deux mesures traditionnelles de la recherche d'information dans la tache d'évaluation des performances des systèmes de segmentation [Salton et al., 1996]. La Figure 8.12 ci-dessous présente les valeurs de rappel et de précision pour les cinq textes segmentés par le groupe des sept lecteurs. Cette figure montre que pour tous les lecteurs, les valeurs de précision sont très hautes. Dans le cas du rappel, le taux d'erreur des juges considérés individuellement est aussi médiocre que celui des algorithmes. Cette observation est à mettre en relation avec la faible fiabilité des évaluations de chaque juge. L'analyse de leurs réponses indique que chaque juge, pris individuellement, est peu fiable contrairement à l'indice global de segmentation, qui peut être dérivé des réponses de l'ensemble des juges. Il faut toutefois noter que les juges, pris individuellement, sont eux-mêmes inefficaces dans l'identification des changements de thème.

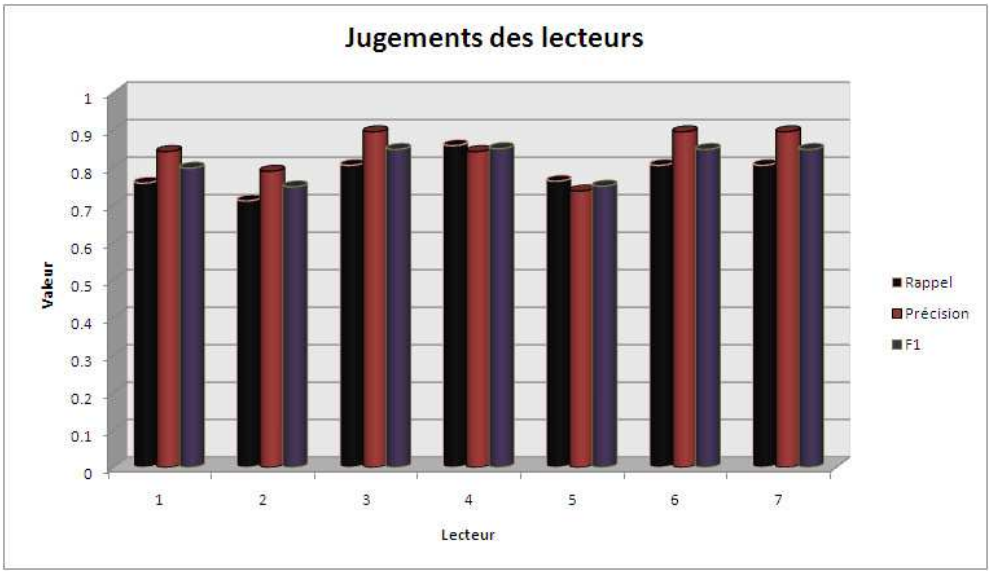


**Fig. 8.12.** Les résultats d'évaluation des jugements des lecteurs.

Le Tableau 8.2 et la Figure 8.13 présentent les valeurs de rappel et de précision pour les cinq textes du corpus de référence segmentés par l'algorithme TextTiling. TextTiling a obtenu ses

meilleures valeurs pour le premier texte. La valeur de la mesure de rappel pour le texte 1 est de 0.75 alors que la valeur de la mesure de précision est de 0.95.

| Texte | Nombre total de frontières | Nombre de frontières en accords | Rappel | Précision |
|---|---|---|---|---|
| 1 | 6 | 6 | 0.75 | 0.95 |
| 2 | 4 | 4 | 0.66 | 0.80 |
| 3 | 3 | 2 | 0.57 | 0.66 |
| 4 | 5 | 2 | 0.37 | 0.77 |
| 5 | 1 | 1 | 0.40 | 0.89 |

**Tableau 8.2.** Rappel et Précision pour 5 textes segmentés avec l'algorithme *TextTiling.*

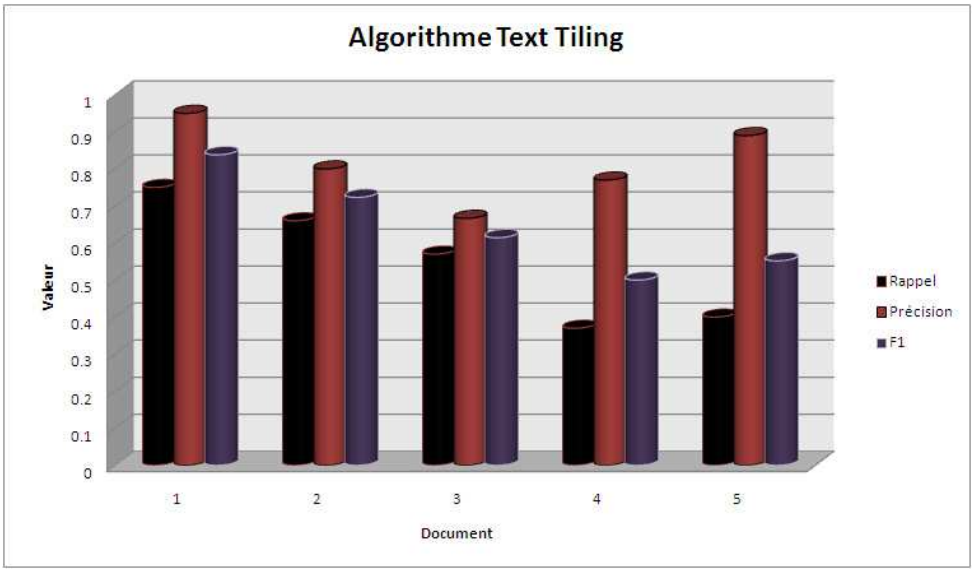


**Fig. 8.13.** Les résultats d'évaluation pour l'algorithme *TextTiling*.

Le Tableau 8.3 et la Figure 8.14 présentent les valeurs de rappel et de précision pour les cinq textes segmentés avec l'algorithme C99. Nous remarquons que les valeurs obtenues par l'algorithme C99 sont en général des valeurs moyennes. La plus haute valeur de rappel est de 0.92 et la plus haute valeur de précision est de 0.97, tous les deux obtenues pour Texte 5.

| Texte | Nombre total de frontières | Nombre de frontières en accords | Rappel | Précision |
|---|---|---|---|---|
| 1 | 6 | 3 | 0.45 | 0.50 |
| 2 | 4 | 2 | 0.40 | 0.50 |
| 3 | 3 | 2 | 0.33 | 0.66 |
| 4 | 5 | 3 | 0.40 | 0.60 |
| 5 | 1 | 0 | 0.92 | 0.97 |

**Tableau 8.3.** Rappel et Précision pour 5 textes segmentés avec l'algorithme C99.

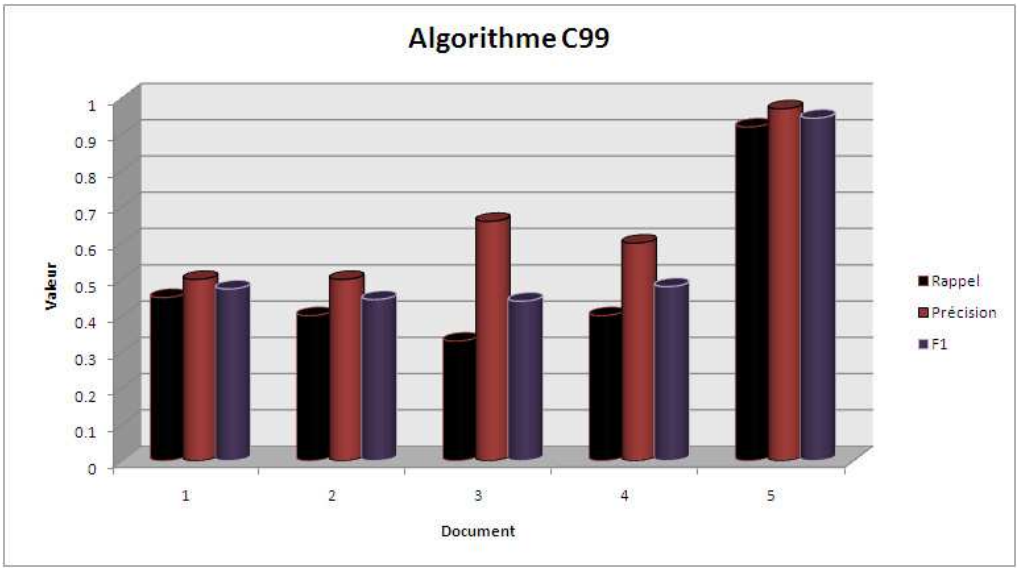


**Fig. 8.14.** Les résultats d'évaluation pour l'algorithme *C99*.

Le Tableau 8.4 présente les résultats de comparaison des deux algorithmes par rapport aux jugements des lecteurs. L'algorithme TextTiling de bonnes valeurs pour toutes les mesures. En précision; il a eu la valeur 0.81 et en rappel il a eu la valeur 0.55. L'algorithme C99 a eu la valeur 0.64 en précision et la valeur 0.55 pour le rappel. Il est nécessaire de noter que la remarque la plus importante pour la segmentation de ce corpus est que la différence entre TextTiling et C99 est statistiquement significatif pour la mesure de précision (+0.17) et pour la mesure $F_1$ (+0.7).

| Segmentation | Rappel | Précision | F1 |
|---|---|---|---|
| Human Judges | 0.81 | 0.84 | 0.82 |
| TextTiling | **0.55** | **0.81** | **0.65** |
| C99 | 0.54 | 0.64 | 0.58 |

**Tableau 8.4.** Comparaison des algorithmes avec les jugements des lecteurs.

## 6. Discussion

Les algorithmes évalués dans cette étude sont parmi les plus efficaces mis au point dans le domaine de la linguistique informatique pour réaliser ce genre de tâches. Même que certaines études ont montré qu'ils peuvent être dépassés, les gains sont souvent très faibles. D'autre part, il a été montré que les algorithmes ne sont pas moins efficaces que les juges, pris un par un. Les analyses ont montré que, dans la plupart des cas, que les juges ne sont pas d'accord entre eux. Par conséquent, il semble que les juges n'avaient pas réalisé la tâche de segmentation avec la rigueur souhaitable. D'autre part, si les deux algorithmes testés sont capables de trouver le début des textes quand ceux-ci sont enchaînés, ils échouent dans la détection des changements de thèmes perçus par la majorité des juges. Contrairement à la fiabilité individuelle d'un juge, la fiabilité globale de l'ensemble des juges est très bonne (la précision moyenne de 0,84). La raison possible de la faiblesse des performances individuelle est liée aux textes utilisés dans cette expérimentation. Contrairement à la plupart des travaux dans ce domaine, nous avons utilisé un corpus de textes en langue arabe, l'une des langues les plus difficiles du point de vue morphologie. La structure thématique des textes analysés est moins explicite, ce qui ajoute une autre dimension de difficulté Seule une analyse fine de la structure linguistique des textes va permettrait donc de résoudre ce problème.

## 7. Conclusion

Dans ce chapitre, une analyse comparative de deux algorithmes de segmentation thématique a été présentée pour un corpus de textes arabes, à savoir l'algorithme TextTiling et C99. Pour évaluer les performances de chaque algorithme, chacun d'eux a été appliqué sur un ensemble de textes arabes puis les résultats ont été comparés. Nous avons confirmé dans ce chapitre que la tâche de segmentation est dure à évaluer ce qui est dû à la variation des objectifs. Nous avons expliqué les différents points de chaque algorithme. Le choix d'implémentation de ces algorithmes a été dicté par la robustesse et la faible demande en ressources pour leur mise au point. En plus, leur utilisation dans le contexte des systèmes de recherche d'information donne généralement de bons résultats. La détermination des ruptures se fait dans un contexte local, c'est-à-dire que les deux algorithmes déterminent la rupture en fonction de ce qu'il y a avant et après et pas dans sa globalité. Mais l'utilisation de la distribution *idf* (Inverse Document Frequency) dans la mesure de la similarité ne permet pas de traiter des textes sous forme de flux. De plus, la présence dans le texte de beaucoup de titres, de petits paragraphes engendrent de mauvais résultats pour l'algorithme TextTiling, car celui-ci ne prend pas en compte l'organisation hiérarchique. Globalement l'algorithme TextTiling paraît être plus adapté à la langue arabe que celui de C99. Pour aller plus loin dans les expérimentations, nous devrions essayer un nouvel algorithme qui mélange des méthodes supervisées avec d'autres méthodes non supervisées, et faire de nouvelles comparaisons entre les approches statistiques et les approches linguistiques. Finalement, notre travail montre qu'avec seulement de petites améliorations, les algorithmes existants pour segmenter des textes anglais, sont adaptables pour les textes arabes. Dans le prochain chapitre, nous exposerons les résultats d'évaluation de ces algorithmes dans le but d'amélioration des performances d'un système de recherche d'information des textes arabes.