VOILADIS : des relations lexicales aux structures de discours

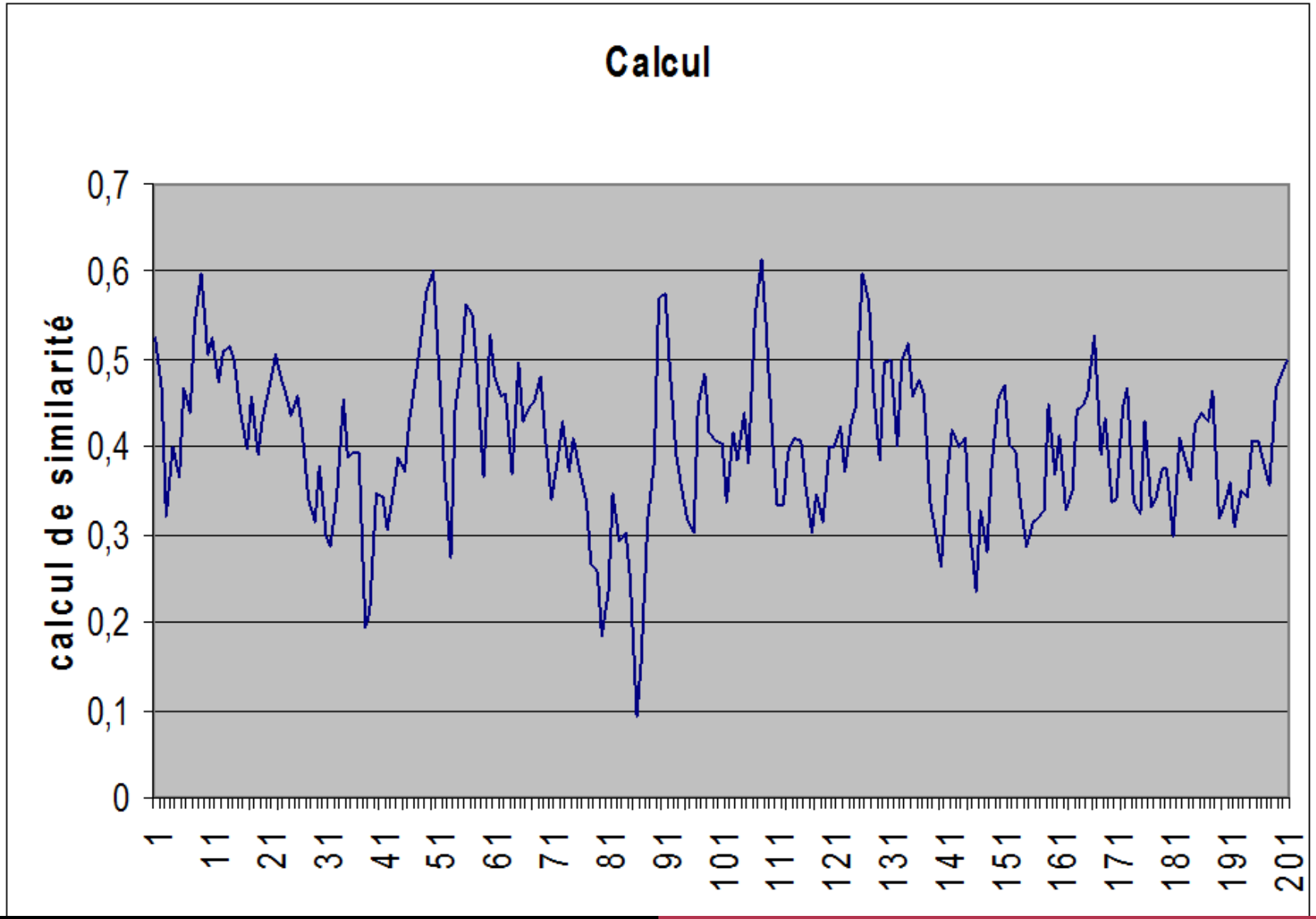
Clémentine Adam, Cécile Fabre, Philippe Muller
CLLE/IRIT

16 Décembre 2008

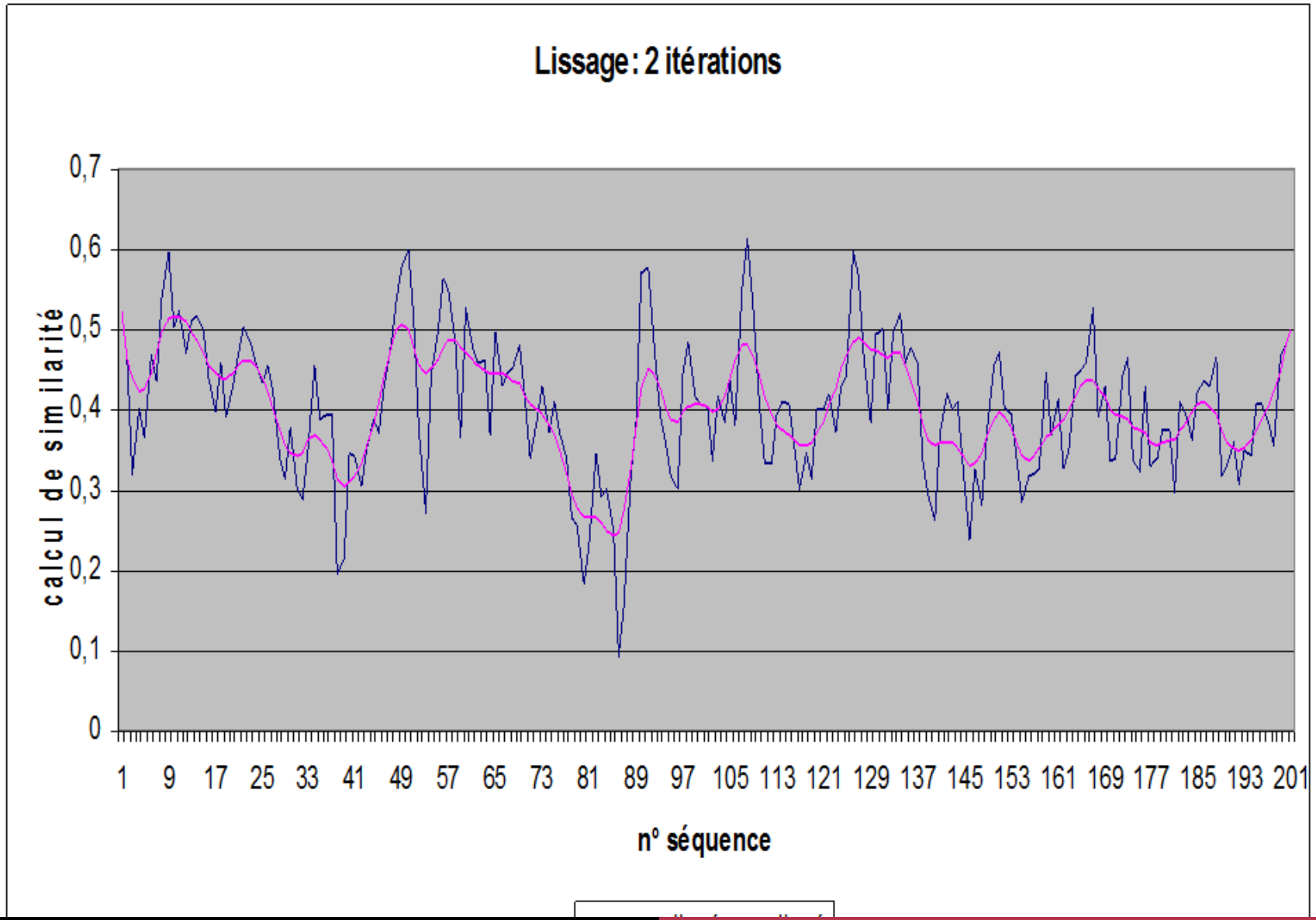
Hearst

- Segmentation linéaire
- Indice utilisé : la réitération lexicale. Pas de ressources extérieures.
- Algorithme :
  - Segmentation du texte en blocs élémentaires de taille arbitraire (20 tokens)
  - Fenêtre de calcul glissante (6 blocs). Pour chaque frontière entre blocs, on calcule un score de similarité basé sur la réitération lexicale (nombre de tokens partagés par les trois blocs précédant la frontière et les trois blocs suivant la frontière)
  - La courbe obtenue est lissée
  - ...

Hearst


Calcul
calcul de similarité
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201

Hearst


Lissage : 2 itérations
calcul de similarité
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145 153 161 169 177 185 193 201
n° séquence

Hearst

Algorithme :

- Segmentation du texte en blocs élémentaires de taille arbitraire (20 tokens)
- Fenêtre de calcul glissante (6 blocs). Pour chaque frontière entre blocs, on calcule un score de similarité basé sur la réitération lexicale (nombre de tokens partagés par les trois blocs précédant la frontière et les trois blocs suivant la frontière)
- La courbe obtenue est lissée
- Toutes les profondeurs des vallées sont calculées
- Celles qui sont supérieures à l'écart-type sont considérées comme correspondant aux ruptures du texte
- Les ruptures sont ajustées au paragraphe

Kozima 1993

- Segmentation linéaire
- Algorithme similaire à celui de Hearst
- Apport : utilisation d'une ressource lexicale générique : scores de similarités entre mots calculés à partir de leurs cooccurrences dans les entrées d'un dictionnaire.
- La similarité entre deux séquences est calculée en se basant sur les similarités des mots qui les composent

Choi 2000,2001

- Segmentation "matricielle" : la proximité entre chaque paire de phrase est calculée
- Algorithme :
  - Segmentation en phrases
  - Calcul de similarités entre chaque paire de phrases - création d'une matrice
  - Création d'une nouvelle matrice (*rank matrix*) en remplaçant chaque score par son rang dans une fenêtre locale
  - ...

Choi 2000,2001

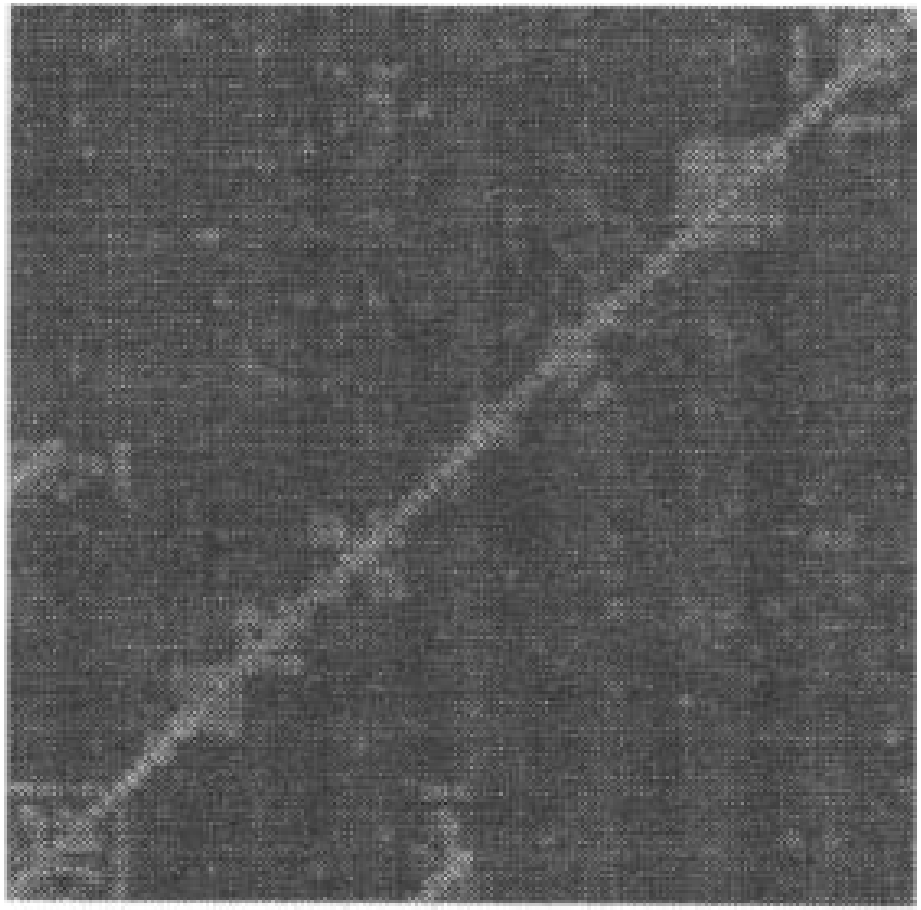


Figure 1: An example similarity matrix.

Choi 2000,2001

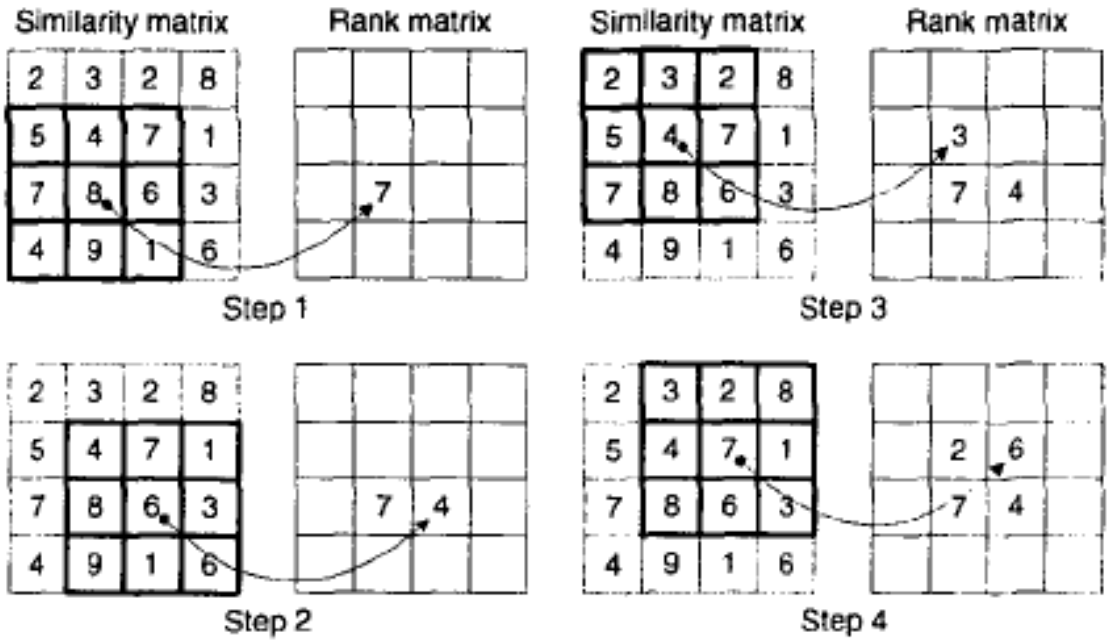


Figure 2: A working example of image ranking.

Choi 2000,2001

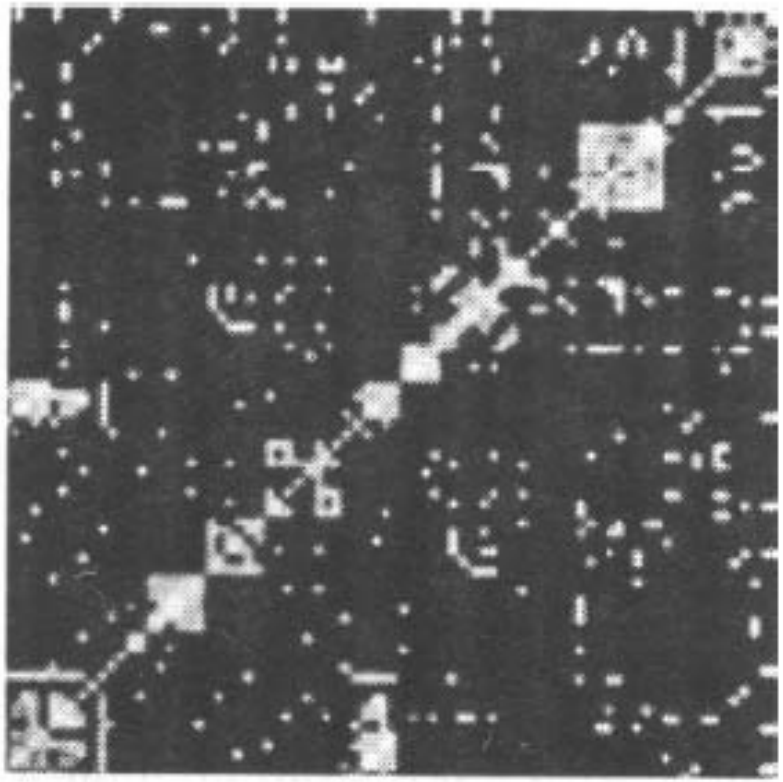


Figure 3: The matrix in figure 1 after ranking.

Choi 2000,2001

- Algorithme :
  - Segmentation en phrases
  - Calcul de similarités entre chaque paire de phrases - création d'une matrice
  - Création d'une nouvelle matrice (*rank matrix*) en remplaçant chaque score par son rang dans une fenêtre locale
  - Le document est segmenté selon les frontières entre unités qui maximisent la somme des similarités moyennes à l'intérieur des segments constitués
- Principal apport : utilisation de l'analyse sémantique latente (LSA) : Des proximités entre les mots sont calculées en corpus (dans un corpus qui contient le texte à segmenter). Ces proximités sont basées sur les cooccurrences des mots à l'intérieur des mêmes textes / paragraphes / phrases. Elles sont utilisées lors du calcul de similarité entre les phrases.

Évaluation : Quelle référence utiliser ?

- Assemblage de textes différents
  Inconvénients : pas un vrai texte mais un objet fabriqué !
  Circularité : on crée ce qu'on postule...
- Marquage manuel ?
  Inconvénients : l'accord inter-annotateur est souvent très faible !
- Marquage typographique (titre, paragraphe) ?
  Inconvénients : biaisé ?

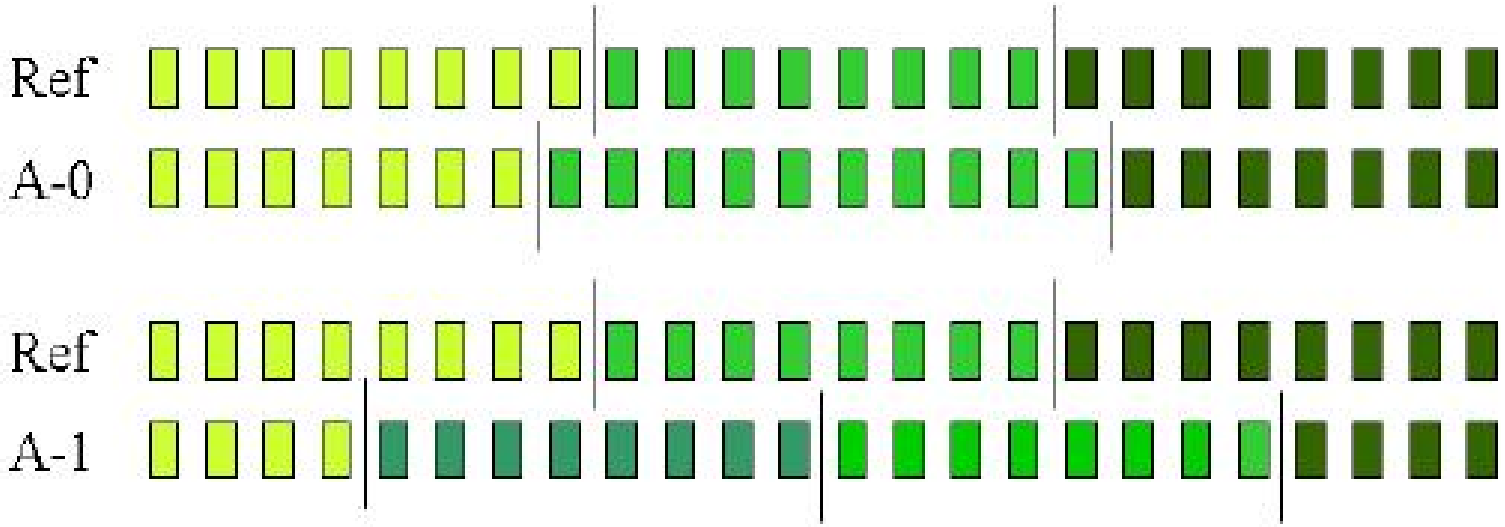
Évaluation : Quelle mesure utiliser ?

- précision/rappel des frontières (taux de frontières correctes, taux de frontières trouvées)
- P_k : nb de fois où deux mots pris au hasard à une distance k sont dans le même segment (ou non) à la fois dans la référence et l'hypothèse

$$P_k = \frac{1}{n-k} \sum_{i \in [0, n-k]} (\delta_{ref}(i, i+k) = \delta_{hyp}(i, i+k))$$

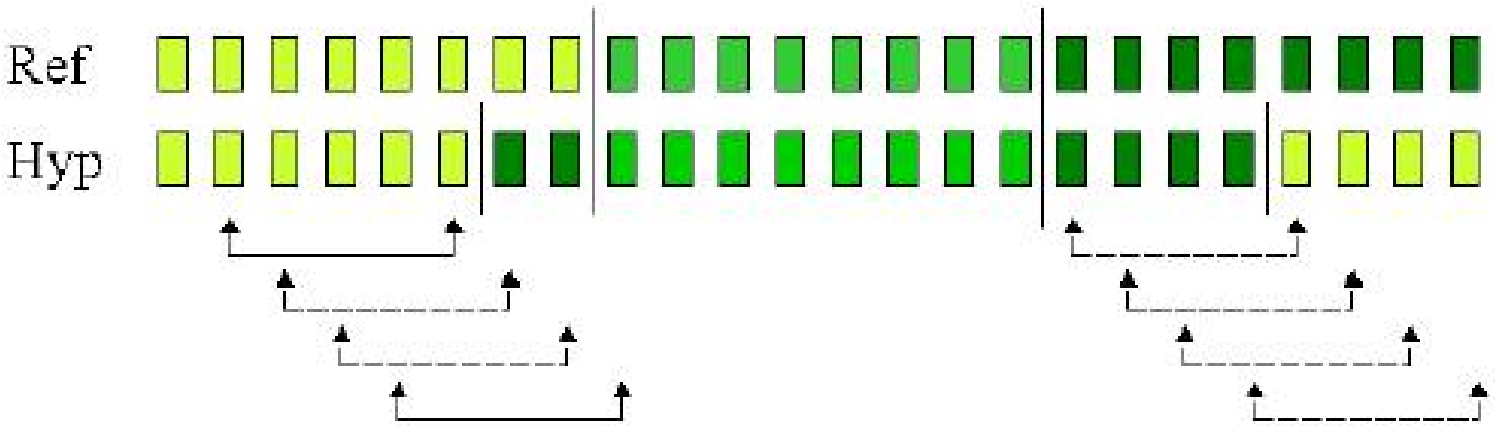
- Windowdiff = différences du nombre de bornes dans une fenêtre glissante
- On pourrait en imaginer d'autres : distance d'édition ? utilisée pour mesurer la distance entre deux chaînes de caractères. Nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre.

Illustration



P_k

fenêtre glissante, trait droit = correct



Utilisation des voisins distributionnels

- Problématique : quel peut-être l'apport de l'utilisation de cette ressource ?
- Calculés en corpus, à partir de Wikipedia
- Une ressource importante, à explorer... Comment les filtrer, comme les faire intervenir dans la segmentation ?

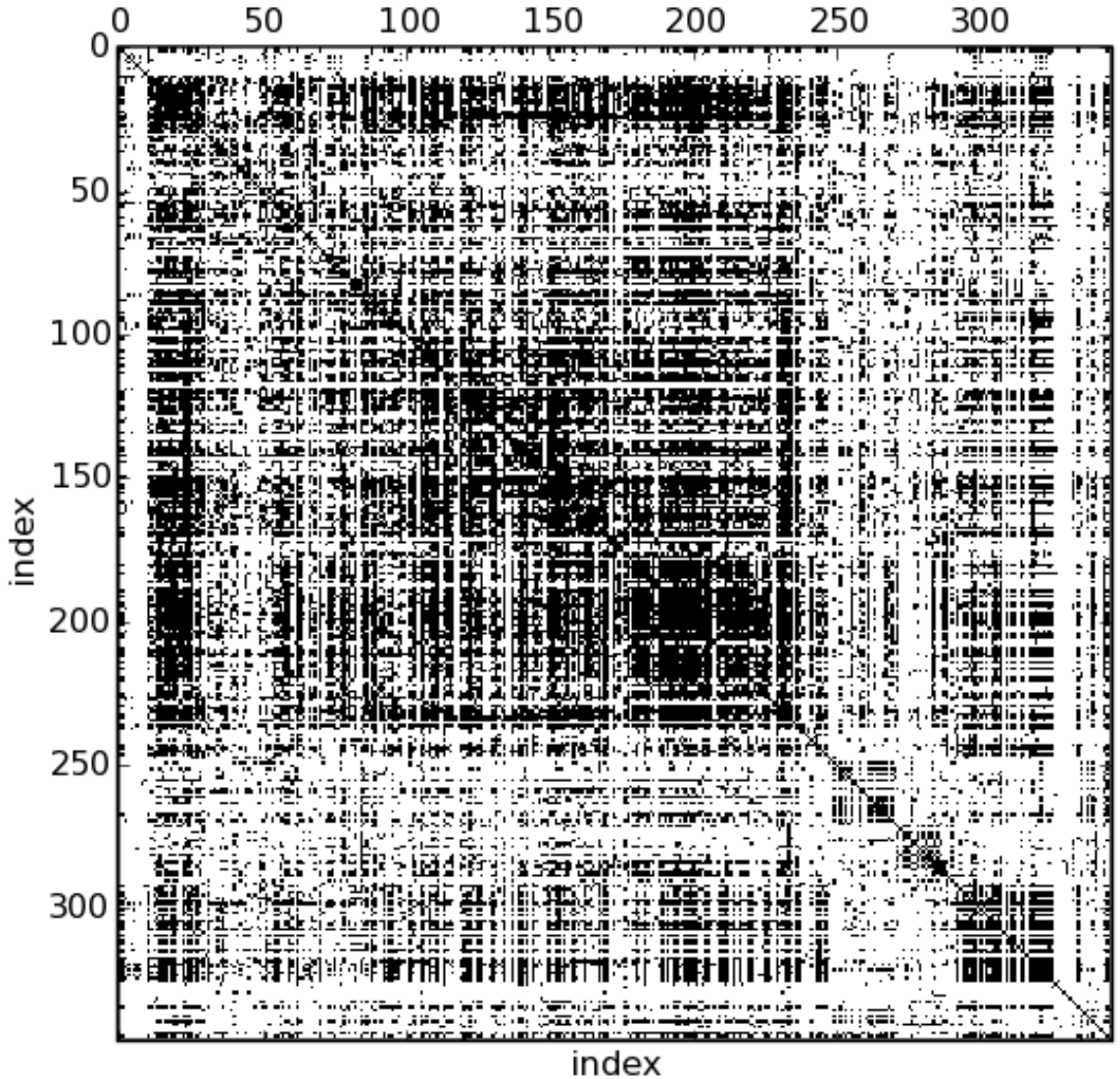
Reprise de Hearst

- Application de l'algorithme de Hearst
- Segmentation et lemmatisation des textes avec le TreeTagger
- Dans le premier cas : uniquement réitération lexicale (à partir des lemmes)
- Dans le second cas : utilisation des voisins
- Évaluation en comparant avec les coupures des titres

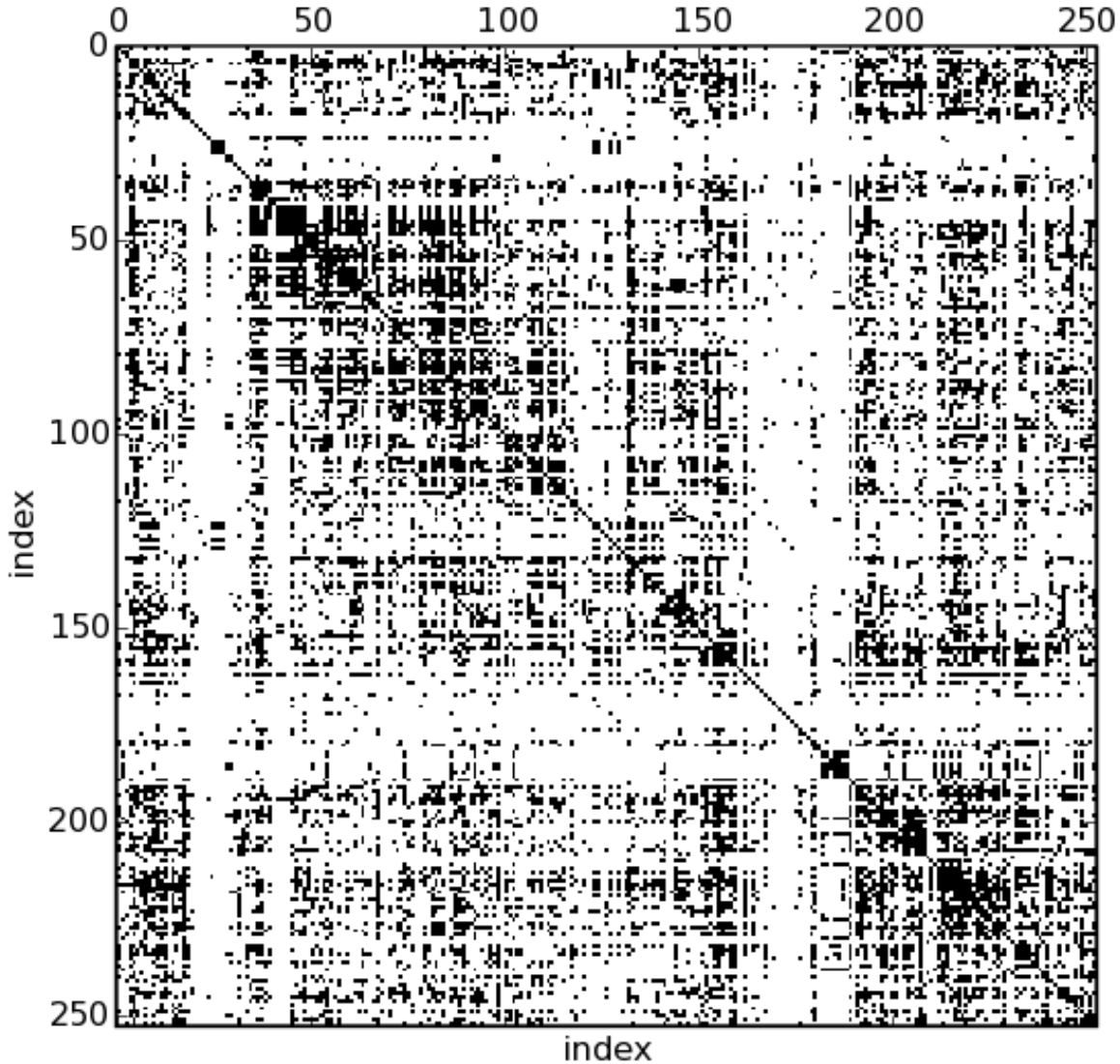
Résultats

	Réit. lex.	Voisins
pr	0.33	0.26
rap	0.32	0.36
pk	0.34	0.53
wd	0.36	0.58
edit	4.5	5.5

Visualisations matricielles



Visualisations matricielles



Limites techniques du TextTiling

- peu de données
- nombre de segments souvent présupposé
- ou bien benchmarks artificiels (textes concaténés)
- la plupart des approches prennent des décisions très locales
- tâche de segmentation mal définie : accord inter-annotateurs assez mauvais

Limites théoriques du TextTiling

Widlocher 2008:40-41

- "L'idée d'une structure textuelle conçue comme *séquence de segments contigus* pose problème."
- La notion de thème reste à définir ;
- Plutôt que des unités thématiques, on peut vouloir mettre au jour des unités correspondant à des buts communicatifs ;
- Pavage complet ou segmentation plus sporadique "Ce choix de procéder à un pavage complet [...] est lourd de conséquences théoriques [...] l'idée de traitement *robuste* l'emportant sur une identification fine d'objets linguistiques mieux délimités conceptuellement et dont la signification pourrait être envisagée."