Modèles thématiques pour la segmentation de documents

Hemant Misra [1], François Yvon [2]

[1] Dpt of Computing Science, University of Glasgow - Glasgow - United Kingdom

[2] Université Paris Sud 11 et LIMSI/CNRS – BP 133 – 91403 Orsay cédex – France

Résumé

Dans ce travail, nous considérons la tâche de segmentation automatique de document, envisagée ici avec les des modèles thématiques probabilistes, en particulier l'allocation Dirichlet latente (LDA). Un intérêt de cette approche est qu'elle fournit non seulement une segmentation, mais également une coloration thématique des segments extraits, ce qui est une information potentiellement utile pour des traitements ultérieurs. Son principal défaut est la complexité algorithmique de la méthode. En nous appuyant sur une analyse de l'algorithme de programmation dynamique utilisé pour réaliser la segmentation, nous proposons une méthode de segmentation heuristique qui accélère de manière spectaculaire le temps de traitement, sans dégradation des performances. Notre approche s'avère finalement meilleure qu'une approche de l'état de l'art sur des jeux de données, et surpasse la plupart des approches proposées dans la littérature sur un jeu d'essai standard.

Abstract

In this work, we consider the text segmentation task from a topic modeling perspective, and introduce a novel application of the Latent Dirichlet Allocation (LDA) topic model: to segment a text into semantically coherent blocs. A major benefit of the proposed approach is that along with the segment boundaries it outputs the topic distribution associated with each segment, which is of potential use in applications like discourse analysis. However, a drawback of our approach is its unrealistically high computational cost. Based on an analysis of the dynamic programming algorithm used for segmentation, we suggest an optimization that dramatically speeds up the process, with no loss in performance. The new approach outperforms a standard baseline method on two databases and yields better performance than most of the available methods on a benchmark database.

Keywords: text segmentation, topic modeling, latent Dirichlet allocation, dynamic programming

1. Introduction

La segmentation automatique de documents consiste à proposer un découpage d'un document en fragments thématiquement homogènes. Cette tâche est particulièrement utile pour des applications de recherche d'information, dans lesquelles il est parfois préférable d'indexer et de présenter aux utilisateurs des portions d'un document plutôt que le document entier, surtout s'il est long. Il en va de même pour la recherche dans des documents audio-visuels (radio- ou télé-diffusés), qu'il est souhaitable de consulter en accédant directement aux segments pertinents, plutôt que d'avoir à parcourir linéairement l'intégralité du document. Segmenter est enfin utile pour produire automatiquement des résumés par extraction : disposer d'une segmentation permet de n'extraire une ou deux phrases par segment.

La méthode la plus répandue pour effectuer cette segmentation est d'utiliser des techniques d'apprentissage non-supervisé, qui s'appuient principalement sur l'hypothèse que les mots ou

lemmes tendent à être répétés au sein des fragments qui sont thématiquement cohérents, et que des changements lexicaux marquent des frontières de segments (Hearst, 1997 ; Reynart, 1998 ; Choi, 2000). Ces techniques présentent l'avantage de ne demander aucune forme d'apprentissage et de s'appliquer directement à n'importe quel type de texte, au prix de prétraitements minimaux (segmentation en mots, optionnellement suivie d'une lemmatisation/racinisation). Pour pallier les problèmes dus à la variabilité morphologique des mots-formes ou aux emplois de synonymes ou de termes sémantiquement apparentés, il est possible d'étendre cette méthodologie en utilisant des représentations sémantiques latentes, permettant d'obtenir des performances améliorées (Choi et al., 2001 ; Brants et al., 2002).

Dans ce travail, nous nous intéressons à l'utilisation d'une autre famille de techniques s'appuyant sur des modélisations probabilistes des thèmes (Nigam et al., 2000, Blei et al., 2002): l'hypothèse ici est qu'il est possible de détecter des thèmes latents au sein d'un document, la connaissance desquels permet, dans un second temps, de segmenter le document. Ces techniques diffèrent des précédentes en ceci (i) qu'elles demandent une étape préalable d'estimation des modèles, ce qui pose la question de la disponibilité de données d'apprentissage adaptées et (ii) qu'elles fournissent non seulement une segmentation, mais également une coloration thématique des passages extraits, qui peut être utilisée dans des phases ultérieures de traitement. Un second défi que ces méthodes posent est, nous le verrons, d'ordre computationnel. La contribution de ce travail est donc double : nous montrons d'une part, que, modulo la disponibilité d'un corpus d'apprentissage adapté, nos méthodes s'avèrent meilleures que l'état de l'art, incarné ici par les propositions d'Utiyama et Isahara (2001); nous introduisons également une heuristique, qui permet de rendre ces traitements efficaces.

Cet article est organisé comme suit : dans la section 2, nous rappelons brièvement les principales caractéristiques des modèles probabilistes considérés dans cette étude; en section 3 nous décrivons les algorithmes de programmation dynamique qui sont utilisés dans les applications de segmentation. La section 4 présente le protocole expérimental, ainsi que les résultats obtenus sur plusieurs corpus. Nous introduisons, en section 5, une heuristique qui permet d'accélérer très significativement la segmentation thématique. Des conclusions et diverses perspectives pour prolonger cette étude sont finalement discutées en section 6.

2. Modèles probabilistes de thèmes : une brève introduction

Les modèles probabilistes considérés dans cette étude représentent les documents comme des « sac-de-mots », plus précisément comme des vecteurs d'occurrences indexés par les mots. Ils permettent de construire, de manière non supervisée, des partitionnements non-déterministes d'un ensemble de documents, qui associent à chaque document un vecteur donnant sa probabilité dans chacun des thèmes identifiés par la méthode. Nous notons n_D, n_W, n_T respectivement le nombre de documents, la taille du vocabulaire et le nombre de thèmes. $C_{w,d}$ est le terme général du vecteur de comptes C_d et désigne le nombre d'occurrence du mot w dans le document d ; l_d dénote la longueur (le nombre d'occurrences) de d.

2.1. Le mélange de multinomiales

Le mélange de multinomiales (Nigam et al., 2000 ; Rigouste et al., 2007) modélise le vecteur de comptes par une loi de mélange : on suppose que les documents du corpus se ventilent dans n_T thèmes; chaque thème t a une probabilité *a priori* α_t et est associé à une distribution multinomiale paramétrée par le vecteur ϕ_t. La vraisemblance d'un document d est :

$$P(d;\alpha,\phi) = \sum_t P(t;\alpha,\phi)P(C_d \mid t;\alpha,\phi) = \sum_t \alpha_t \frac{l_d!}{\prod_w C_{w,d}!} \prod_w \phi_{w,t}^{C_{w,d}} \quad (1)$$

Cette expression rend compte du modèle de génération sous-jacent pour un document : choix d'un thème *t* avec la distribution paramétrée par α, puis, conditionnellement au thème, génération des occurrences par tirage multinomial. La vraisemblance du corpus complet est un produit de tels termes. L'estimation du modèle consiste à obtenir les valeurs du vecteur de paramètres α et des n_T vecteurs ϕ_t. La maximisation de la log-vraisemblance du corpus n'a pas de solution analytique, mais peut être résolue par des procédures itératives telles que l'algorithme EM (qui est la méthode que nous avons utilisée) ou encore par échantillonnage de Gibbs. L'utilisation d'un hyper-paramètre β, qui peut s'interprète comme le paramètre unique d'une loi de Dirichlet définissant un terme d'a priori sur le vecteur ϕ_t permet de « lisser » ces estimateurs et d'éviter les valeurs nulles. On se reportera aux références citées plus haut pour des détails sur ces procédures. Une fois les paramètres connus, la vraisemblance d'un document se calcule par application directe de la formule (1) ci-dessus.

2.2. Allocation Dirichlet Latente

Le modèle LDA (pour *Latent Dirichlet Allocation*) introduit dans (Blei et al., 2002 ; Griffiths and Steyvers, 2004) vise à généraliser le modèle précédent, en proposant que l'association entre thèmes et documents soit médiatisée par les occurrences. Dans ce modèle, ce sont donc les occurrences, plutôt que les documents, qui sont associées à des thèmes. Les différentes occurrences au sein d'un même document restent toutefois liées par une variable latente qui contrôle globalement la distribution des thèmes au sein du document.

Le modèle de génération complet associé à ce modèle est le suivant:

- Pour chaque thème, tirer, selon une loi de Dirichlet de paramètre β, les paramètres $\phi_t = (\phi_{1,t}...\phi_{n_W,t})$ des lois discrètes qui probabilisent les occurrences des mots du vocabulaire. Comme précédemment, $\phi_{w,t}$ s'interprète comme la probabilité de l'occurrence du mot *w* dans un document du thème *t*.
- Pour chaque document $d \in \{1...n_D\}$
  - Tirer le vecteur θ_d représentant la distribution des thèmes dans *d* selon une loi de Dirichlet de paramètre α. Chaque $\theta_{d,t}$ désigne donc la proportion des occurrences du document *d* qui sont associées au thème *t*.
  - Pour chaque position *i* dans *d*, *i* variant de 1 à l_d:
    - Choisir le thème t_i associé à l'occurrence *i* selon θ_d.
    - Choisir le mot *w* à la position *i* selon ϕ_{w,t_i}

Le choix d'un thème est donc effectué *indépendamment* pour chaque occurrence du document, sous la contrainte du respect global de la distribution des thèmes fixée par θ_d : il est donc possible de considérer des changements de thème à chaque position du document.

L'estimation du modèle est difficile, du fait de la forme particulière de la log-vraisemblance du corpus, qui s'exprime comme une intégrale, par rapport aux variables latentes θ_d, de termes $P(d \mid \theta_d;\phi,\alpha,\beta)$ (voir l'équation (3) ci-dessous). La log-vraisemblance ne peut donc être optimisée directement. Des procédures d'estimation ont toutefois été proposées par (Blei et al., 2002), ainsi que par (Griffiths and Steyvers, 2004) : comme ces auteurs, nous utilisons pour les paramètres $\phi_{w,t}$ et $\theta_{t,d}$ des estimateurs obtenus par échantillonnage de Gibbs (voir également Rigouste et al., 2006).

L'inférence, qui consiste à déterminer la distribution des thèmes pour des documents de test, est également difficile. Nous suivons ici la proposition de (Heidel et al., 2007; Misra et al., 2008) qui consiste à utiliser une procédure itérative reposant sur la mise à jour suivante :

$$\theta_{t,d} \leftarrow \frac{1}{l_d} \sum_{w} \frac{C_{w,d} \theta_{t,d} \phi_{w,t}}{\sum_{t'} \theta_{t',d} \phi_{w,t'}} \tag{2}$$

Une fois la distribution de thèmes θ_d connue, la log-vraisemblance d'un document de test s'en déduit directement selon :

$$\log(P(d \mid \theta; \phi, \alpha, \beta)) = \sum_{i=1}^{l_d} \log(\sum_{t} \theta_{d,t} \phi_{w_i,t}) \tag{3}$$

Retenons simplement de cette très brève présentation que le calcul de la vraisemblance d'un document de test implique la connaissance de la distribution θ_d et que ce calcul demande d'itérer jusqu'à convergence de la règle de mise à jour (2) énoncée ci-dessus.

3. Segmenter par programmation dynamique

3.1. Le principe général et son implantation par Utiyama et Isahara

3.1.1. Segmenter : un problème de chemin

La tâche de segmentation se formalise comme l'identification d'un ensemble optimal, de taille inconnue, de n_S segments (b, e), où b et e dénotent respectivement le début et la fin d'un segment et correspondent à des indices de phrase, avec $b_{i+1} = e_i$. Il s'agit d'un problème d'optimisation combinatoire, qui demande de considérer toutes les segmentations possibles d'un document. Lorsque la mesure de qualité d'une segmentation se décompose comme une somme de n_S termes, qui chacun se calcule localement, c'est-à-dire en se fondant simplement sur l'analyse du contenu du segment i, ce problème peut être résolu de manière efficace. Fig. 1 représente ce problème d'optimisation sous la forme d'un problème de plus court chemin dans un graphe, où la évaluation d'un arc dépend de « la qualité » du segment correspondant (voir ci-dessous). Par exemple, le coût total de la segmentation (1,3)(4,6) se calcul en sommant les évaluations associées aux arcs correspondants.

Présenté sous cet angle, il est clair que le calcul de la segmentation optimale est résolu efficacement, à l'instar des problèmes de plus courts chemins, par des algorithmes de programmation dynamique.

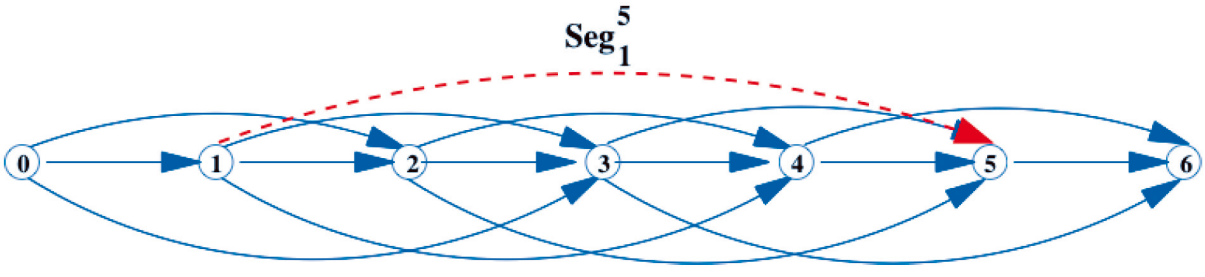

Figure 1: Segmentation optimale par programmation dynamique

3.1.2. Modélisation des segments

L'implémentation qui nous sert de référence, celle de (Utiyama and Isahara, 2001), adopte un modèle probabiliste pour évaluer les segments. Soit d un document contenant n_P phrases $p_1 ... p_{n_P}$, la probabilité d'une segmentation σ de d s'écrit , en utilisant la règle de Bayes:

$$P(\sigma \mid d) = \frac{P(d \mid \sigma)P(\sigma)}{\sum_{\sigma'} P(d \mid \sigma')P(\sigma')}$$

Le calcul de la segmentation optimale ne demande que d'optimiser le numérateur, ce qui implique toutefois de disposer de modèles pour $P(\sigma)$ et pour $P(d \mid \sigma)$. Pour le premier terme, qui définit un a *priori* sur les segmentations, les auteurs s'en tiennent à une expression simple qui rend moins probable les segmentations les plus longues (en nombre de segments), en posant : $\log(P(\sigma)) = -n_S \log(l_d)$. Pour calculer le second terme, les auteurs utilisent un modèle unigramme pour chaque segment σ_i, modèle dont les paramètres sont estimés localement par décompte (avec un lissage de Laplace, introduit par le terme ε pour éviter les problèmes numériques). Si $\sigma_i = (b_i, e_i)$ contient les phrases $p_{b_i} ... p_{e_i}$, on aura alors (on note comme précédemment C_{w,σ_i} le nombre d'occurrences du mot w dans le segment σ_i):

$$P(p_{b_i} ... p_{e_i} \mid \sigma_i) \propto \prod_w \left(\frac{C_{w,\sigma_i} + \varepsilon}{\sum_{w'} (C_{w',\sigma_i} + \varepsilon)}\right)^{C_{w,\sigma_i}} \quad (4)$$

La vraisemblance jointe d'un couple (segmentation, document) s'écrit alors comme un produit de termes similaires au terme précédent, complété d'un terme exprimant la probabilité *a priori* d'une segmentation. Pour trouver la segmentation optimale, il suffit de pondérer chaque arc du graphe précédent d'un coût intégrant le log de la probabilité définie supra en (4) et le terme d'*a priori*, puis de chercher le plus court chemin dans le graphe. Notons que ce problème se résout avec une complexité quadratique en n_P, non compris le calcul des $O(n_P^2)$ évaluations. Dans ce modèle, ce calcul n'augmente pas la complexité totale de l'algorithme et peut être effectué de manière efficace en maintenant les décomptes d'occurrences associés à chaque segment qui se recombinent simplement par sommation lorsque deux segments sont concaténés. Cet algorithme a donc une complexité théorique en $O(n_P^2)$.

3.2. Segmenter avec des modèles de thèmes

L'intuition principale de notre approche est que si les segments recherchés doivent présenter une homogénéité thématique, alors l'utilisation d'une modélisation explicites des thèmes peut s'avérer pertinente. Dans le modèle multinomial (un thème par segment), on s'attend à ce que la vraisemblance du document à segmenter sera d'autant meilleure que les frontières de segments inférées contiennent des mots caractéristiques d'un même thème. L'intuition est identique pour LDA, qui autorise que plusieurs thèmes soit activés dans un même segment.

D'un point de vue conceptuel, l'adaptation de l'algorithme présenté ci-dessus pour des modèles thématiques est sans difficulté et n'implique que de modifier le terme $P(d \mid \sigma)$. Pour le calculer dans le modèle multinomial, il suffit d'utiliser l'expression de la probabilité donnée ci-dessus à l'équation (1) pour le pseudo-document constitué des phrases constituant le segment. Si l'on stocke, pour chaque arc du graphe, un vecteur donnant la probabilité du segment correspondant dans chacun des thèmes, on peut réaliser, comme précédemment, ces calculs de manière incrémentale, et donc sans augmentation de la complexité algorithmique.

Lorsque l'on utilise le modèle LDA, la modification est similaire, à ceci près qu'elle demande, pour chaque segment, de mettre en œuvre la procédure d'inférence itérative décrite à la section 2.2.2. Même en limitant le nombre d'itérations, *ce calcule impose d'examiner plusieurs fois les occurrences d'un segment et ne peut être réalisé de manière incrémentale*. Pour cette variante, la complexité de l'algorithme de segmentation est en principe cubique en le nombre de phrases, ce qui s'avère être un problème pour traiter de longs documents, et justifie l'approche heuristique présentée à la section 5.

4. Expériences

4.1. Bases de données et protocole expérimental

4.1.1. Les corpus

Les expériences décrites ci-dessous utilisent deux corpus. Le premier est régulièrement utilisé, depuis les expériences de (Choi, 2000) pour évaluer les outils de segmentation automatique [1]. Ce corpus artificiel contient 400 pseudo-documents en langue anglaise, artificiellement construits en agrégeant *exactement 10 fragments de textes* extraits du *Brown Corpus* (Francis and Kucera, 1979). Il est divisé en 4 sous-parties, dénotées 3-5, 6-8, 9-11 et 3-11. Un document de la partie 6-8, par exemple, ne contient que des fragments contenant de 6 à 8 lignes. Les documents constitués de courts segments sont plus difficiles à segmenter que ceux pour lesquels on observe plus de phrases, et donc plus de répétitions.

Le second corpus a été dérivé par nos soins du corpus Reuters RCV1 [2] pour simuler une plus grande variété de situations : il contient, comme le corpus de Choi, 4 sous-parties pour les segments courts, plus une cinquième, ci-dessous identifiée par 0-0, qui est construite en concaténant *des documents complets* (et donc de longueur très variable). Par ailleurs, nous avons également pris soin de faire varier le nombre de segments dans un document, en constituant des pseudo-documents à partir de 10, 50 ou 100 articles du corpus Reuters. L'autre intérêt de ce corpus est qu'il laisse disponibles, pour estimer les modèles de thèmes, de nombreux documents qui sont très homogènes avec ceux qui constituent la base de test. Cet apprentissage, aussi bien pour le modèle LDA et le modèle multinomial, a été réalisé à partir d'un sous-ensemble contenant environ 30.000 documents. Pour ces deux modèles, le nombre de thèmes a été fixé arbitrairement à 50. Nous avons également été conduits à reproduire la méthode de Utiyama et Isahara, qui sert de point de comparaison ('Baseline') par la suite.

4.1.2. Mesure des performances

Tous nos résultats expérimentaux sont présentés en utilisant la métrique P_k introduite dans (Beeferman et al., 1999). Intuitivement, cette métrique mesure la probabilité que deux phrases appartenant à un même fragment soient affectées à deux segments par l'algorithme de segmentation. Plus ces valeurs sont faibles, meilleur est l'algorithme de segmentation.

4.2. Expériences avec les données de Choi

Tab. 1 contient les principaux résultats de cette première série d'expériences. Ces résultats semblent sans appel : la méthode de base, celle de Utiyama et Isahara, s'avère à la fois bien

[1] Téléchargeable depuis http://www.freddychoi.co.uk/.
[2] http://about.reuters.com/researchandstandards/corpus/.

meilleure et bien plus rapide que l'utilisation de modèles de thèmes, et ne demande, de surcroît, aucun apprentissage. L'utilisation d'un raciniseur (stemming) permet d'en améliorer encore un peu la précision. Elle semble donc bien meilleure que celle que nous proposons. Notons également que l'approche fondée sur LDA est meilleure, et comme prévu, bien plus lente que celle qui utilise le modèle de mélange de multinomiales. Enfin, comme attendu, les performances s'améliorent pour tous les modèles avec la longueur des segments.

	3-5		6-8		9-11		3-11	
Baseline, avec stemming [3]	13		6		6		11	
Baseline, sans stemming	14	(0,15)	7	(0,65)	7	(1,65)	11	(0,69)
Modèle Multinomial	38	(9,6)	34	(22,3)	34	(46,5)	33	(22,2)
Modèle LDA	22,5	(119,8)	15,4	(580,8)	13,1	(1470,9)	15,5	(566,0)

Tableau 1 : Précision (P_k) et vitesse [4] (en s) des segmenteurs sur les données de Choi

Un examen plus détaillé du comportement de LDA permet d'expliquer cet échec : le corpus de Choi s'avère en fait très différent du corpus Reuters utilisé pour estimer le modèle, ce qui a deux impacts négatifs sur le comportement de l'algorithme : (i) de nombreuses formes (totalisant environ 10% des occurrences) du corpus de test ne sont pas rencontrées dans le corpus d'apprentissage et sont donc simplement ignorées par notre méthode ; (ii) le corpus de test s'avère également moins diversifié que notre corpus d'apprentissage, ce qui a pour conséquence que près de 50% des segments se retrouvent assignés majoritairement à un seul et même thème. Difficile dans ces conditions, de prévoir précisément les frontières de thèmes.

Pour vérifier que cette analyse est correcte, nous avons réitéré les mêmes expériences, en subdivisant le corpus de Choi en deux sous-ensembles de 200 pseudo-documents. Le premier (part A) est intégré au corpus utilisé pour estimer les modèles thématiques, le second est utilisé pour les tests (part B). Nous espérons ainsi réduire la différence entre corpus d'apprentissage et corpus de test qui semble fortement dégrader la précision de notre méthode. Les résultats expérimentaux sont rassemblés dans Tab. 2, et donnent une toute autre image des mérites relatifs des différents modèles. En adaptant les modèles de thèmes, nous obtenons des performances bien meilleures que celles du modèle de base (en fait bien meilleures que toutes celles rapportées dans la littérature jusqu'à présent pour ce jeu de données, en particulier celles de (Choi et al., 2001 ; Brants et al., 2002, Fragkou et al., 2004).

	3-5	6-8	9-11	3-11
Baseline	14,9	8,1	7,7	11,2
Multinomial	39,1	34,9	34,5	31,5
Multinomial + adaptation	**1,6**	**1,2**	**2,1**	**1,4**
LDA	23,0	15,8	14,4	15,5
LDA + adaptation	**2,2**	**2,3**	**4,1**	**2,3**

Tableau 2 : Précision (P_k) des segmenteurs sur les données de Choi (part B)

Le modèle multinomial s'avère un peu meilleur que LDA, probablement du fait de la faible diversité des documents du corpus de Choi, qui sont bien modélisés par ce modèle très simple.

[3] Ces résultats sont repris de la publication originale de Utiyama et Isahara.

[4] Ici et ailleurs, ces mesures de vitesse sont effectuées sur une machine de bureau équipée de 4 cœurs, chacun étant cadencé à 3Gz, et ne valent que pour les ordres de grandeur relatifs des différentes méthodes. Elles correspondent au temps moyen pour segmenter un document de test.

4.2. Expériences avec les données Reuters

L'utilisation du jeu de test dérivé du corpus Reuters permet d'évaluer ces différents modèles dans des conditions plus diverses. Les résultats complets sont détaillés dans Tab. 3.

	3-5			6-8			9-11			3-11			0-0		
Baseline	16	36	41	14	34	41	14	33	40	14	34	40	17	29	38
Multinomial	14,8	19,4	21,0	13,3	14,7	16,2	12,1	13,1	14,4	13,8	15,6	16,5	11,5	12,1	13,4
LDA	6,5	9,6*	??	4,5	5,8*	??	5,6	??	??	5,9	??	??	11,1	??	??

Tableau 3 : Précision (P_k) des segmenteurs sur le jeu de test Reuters.
Chaque cellule contient 3 valeurs, correspondant aux trois conditions 10, 50 et 100 fragments par document. Les résultats avec (*) sont des résultats partiels, les résultats manquants sont remplacés par des ??

Ce tableau appelle un certain nombre de commentaires. En premier lieu, le modèle de base se montre sous un jour nettement moins favorable que sur le corpus précédent, ses performances se dégradant très fortement lorsque l'on s'intéresse à des documents plus longs. Les performances des modèles thématiques sont globalement meilleures et moins dépendantes du nombre de segments dans un document. Pour les expériences que nous avons pu conduire à leur terme, le segmenter basé sur LDA semble meilleur que celui utilisant un mélange de multinomiales. Il est cependant considérablement plus lourd à mettre en œuvre que les autres, et n'a pu être utilisé pour segmenter les documents les plus longs, ce qui explique les résultats partiels ou manquants dans Tab. 3. Pour donner un ordre de grandeur du temps de calcul, le test portant sur la condition 3-5, 50 fragments par documents met près de deux heures pour traiter un document de test (soit 150 fois plus longtemps que pour le modèle multinomial). Nos observations sur le comportement du modèle de base confortent celles de (Malioutov and Barzilay, 2006) et peuvent s'expliquer comme suit : l'estimation du modèle unigramme par simple décompte, qui est réalisée dans ce modèle, n'est pas robuste pour des segments courts, qui comprennent peu d'occurrence, et sa robustesse diminue avec la longueur globale du document, qui implique en général des vocabulaires plus étendus. Par comparaison, les modèles thématiques ne cherchent à estimer qu'une poignée de paramètres, qui correspondent à la distribution *a posteriori* des thèmes dans les documents et sont donc moins sensibles à la longueur des documents.

Notons finalement que la condition dans laquelle des documents entiers sont segmentés (colonne 0-0) n'apparaît pas beaucoup plus difficile que les autres, en dépit de la variabilité de la longueur des documents. Sur la base de ces résultats encourageants, nous proposons dans la section suivante une heuristique efficace qui rend possible l'utilisation de LDA même sur de gros documents.

5. Une heuristique efficace pour la segmentation

Comme expliqué en section 2, la complexité principale des modèles thématiques est due à la méthode d'évaluation des segments. Pour LDA en particulier, chacun des n_P^2 segments possibles demande de mettre en œuvre une procédure itérative qui s'avère très coûteuse. Un remède très simple et efficace, adopté par (Malioutov and Barzilay, 2006), consiste à n'évaluer qu'une petite partie des segments, en fixant des seuils maximaux sur leur longueur *a priori* : seuls les segments inférieurs à un certain seuil sont alors évalués. Notre approche est différente, et se fonde sur l'observation suivante : pendant le déroulement de la recherche de la segmentation optimale, la grande majorité des nœuds de type (b) n'est jamais *actif*, ce qui signifie qu'ils ne correspondent jamais au meilleur début possible d'une hypothèse de segment. Rappelons que

chaque hypothèse de segmentation est associée à un pointeur arrière vers l'indice de la phrase qui débute le segment : on constate alors empiriquement que la majorité des nœuds de type (b) ne sont jamais pointés par ces pointeurs arrières.

Cette observation a deux conséquences importante : elle implique, d'une part, qu'il est possible de repérer les candidats qui sont des « bons » débuts de segments *sans qu'il soit nécessaire d'attendre la fin du document* et permet d'envisager d'effectuer *on line* la segmentation de documents arbitrairement longs. Cette piste ne sera pas poursuivie dans le cadre de ce travail. Un autre corollaire, sur lequel s'appuie notre heuristique, est que si (b) est un début de segment actif pour une hypothèse de fin (e) et si (e') une position située en aval de (e) dans le texte, alors il est inutile d'évaluer les segments (b',e') pour toutes les positions (b') situées en amont de (b). Considérons, à titre d'exemple, Fig. 2.

```
0-1                                                  0-1
1-2 0-2                                              1-2 0-2
2-3 1-3 0-3                                          2-3 1-3 0-3
3-4 2-4 1-4 0-4                                      3-4 2-4 1-4 0-4
4-5 3-5 2-5 1-5 0-5                                  4-5 3-5
5-6 4-6 3-6 2-6 1-6 0-6                              5-6 4-6 3-6
...........                                          ...........
(N-3)-(N-2) (N-4)-(N-2) .... 0-(N-2)                 (N-3)-(N-1) (N-4)-(N-2)
(N-2)-(N-1) (N-3)-(N-1) .... 0-(N-1)                 (N-2)-(N-1) (N-3)-(N-1) (N-4)-(N-1)
(N-1)-N (N-2)-N (N-3)-N ............ 0-N             (N-1)-N (N-2)-N (N-3)-N (N-4)-N
```

Figure 2 : Les segments considérés par l'algorithme de programmation dynamique

Sur le côté gauche de la figure sont listés tous les segments qu'il faudra évaluer pour appliquer l'algorithme de programmation dynamique de manière exacte; sur le côté droit, ceux qu'il est nécessaire d'évaluer pour mettre en œuvre notre méthode heuristique. Ainsi, par exemple, si b=3 est un nœud de début actif pour l'indice de fin e=4, alors il est inutile

d'évaluer les segments (0-5), (1-5), etc., ce qui conduit à réduire très fortement le nombre de segments qui sont évalués. Cette heuristique s'applique indépendamment du modèle de thème qui est utilisé, et conduit, pour le modèle LDA, à une diminution du temps de traitement de l'ordre de 95%. Elle permet également d'obtenir les résultats présentés dans Tab. 4, résultats qui confirment nos anticipations : pour le modèle multinomial, les performances sont quasiment celles de la version exacte, et d'un facteur 10 plus rapides à obtenir. Pour LDA, elles permettent d'aboutir à des performances en segmentation très stables pour les différentes conditions de test, quoiqu'encore très coûteuses à obtenir puisqu'avec ce modèle, la segmentation reste environ 50 à 100 plus lente qu'avec le modèle multinomial.

	3-5			6-8			9-11			3-11			0-0		
Multinomial	14,6	18,5	19,8	12,2	13,8	15,3	11,6	12,3	13,7	13,1	14,6	15,5	12,0	12,0	12,9
LDA	6,5	11,2	13,6	4,4	6,0	7,8	5,4	5,5	6,4	5,9	7,3	8,4	11,9	8,4	8,4

Tableau 4 : Précision (P_k) des segmenteurs heuristiques sur le jeu de test Reuters

On remarquera pour finir que cette heuristique s'avère non seulement profitable en termes de vitesse, mais également en termes de performances, puisqu'elle conduit à supprimer des hypothèses de segmentations qui sinon seraient sélectionnées (à tort) par l'algorithme exact.

6. Conclusion

Dans cet article, nous avons étudié la possibilité de mettre en œuvre des modèles probabilistes de thèmes pour une application de segmentation automatique de documents, l'intérêt est de disposer non seulement d'une segmentation thématique, mais également d'une coloration des thématique des différents segments. Sur la base d'expériences conduites sur deux bases de test très différentes, nos principales conclusions sont que les approches qui utilisent des modèles de thèmes doivent être préférées et conduisent à des segmentations plus précises que le modèle de référence, à condition que l'on dispose de données suffisamment semblables aux textes à segmenter sur lesquelles estimer ces modèles. Cette observation ouvre la voie à l'utilisation de modèles thématiques plus riches et à un couplage plus fin entre segmentation et suivi des thèmes. Nous avons également introduit une heuristique très efficace pour réduire le temps de calcul de ces segmentations, sur la base d'un élagage dynamique de l'espace de recherche du segmenteur : pour les deux modèles considérés, cette heuristique nous a permis d'accélérer la segmentation d'un ordre de grandeur. Il reste encore beaucoup à gagner pour pouvoir utiliser LDA dans des conditions réalistes d'utilisation. Une manière de procéder, en cours d'évaluation, consistera à effectuer la recherche en deux temps, en construisant d'abord un treillis de segmentations possibles avec le modèle multinomial, qui sera ensuite réévalué avec LDA. La seconde perspective ouverte par cette heuristique est la possibilité d'effectuer la segmentation en ligne, c'est-à-dire sans devoir attendre de voir la fin du document.

Références

D. Beeferman, Berger A. and Lafferty J. (1999) Statistical models for text segmentation. *Machine Learning*, 31 : 177-210.

Blei D.M., Ng A.Y. and Jordan M.I. (2002) Latent Dirichlet allocation. In Dietterich, T.G., Becker, S. and Ghahramani Z., editors, *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 601-608, Cambridge (MA) : The MIT Press.

Brants T., Chen T. and Tsochantaridis I. (2002). Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the International Conference on Information and Knowledge Management*, pp. 211-218, New York, USA.

Choi F.Y.Y (2000). Advances in domain independant linear text segmentation. In *Proceedings of the Conference of North American Chapter of the ACL*, Seattle, WA.

Choi F., Wiemer-Hastings P. and Moore J. (2001). Latent semantic analysis for text segmentation. In *Proceedings of 6th EMNLP*, pp.109-117.

Fragkou P., Petridis V. and Kehagias A. (2004). A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information System*, 23(2) : 179-197.

Francis W.H. and Kucera A. (1979). *Brown Corpus Manual*. Technical Report, department of Linguistics, Brown University, Providence, Rhode Island.

Griffiths T.L. and Steyvers M. (2004). Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 101 (supl. 1) : 5228-5235.

Griffiths T., Steyvers M., Blei D.M. and Tenenbaum J. (2005). Integrating topics and syntax. In *Proceedings of NIPS, 17*, Vancouver, CA.

Hearst M. (1997). TextTiling: Segmenting texts into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1) : 33-64.

Heidel A., Chang H. and Lee L. (2007). Language model adaptation using latent Dirichlet allocation

and an efficient topic inference algorithm. In *Proceedings of the European Conference on Speech Communication and Technology*, Antwerp, Belgium.

Malioutov I. and Barzilay R. (2006). Minimum cut model for spoken lecture segmentation. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, pp. 25-32.

Misra H., Yvon F. and Cappé O. (2008). Using LDA to detect semantically incoherent documents. In *Proceedings of the Conference on Computational Natural Language Learning*, Manchester, UK, pp. 41-48.

Nigam K., McCallum A., Thrun S. and Mitchell T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning journal*, 39(2/3) : 103-134.

Ponte J.M. and Croft W.B. (1997). Text segmentation by topic. In *Proceedings of the European Conference on Digital Libraries*, pp. 113-125.

Reynar J.C. (1998). *Topic Segmentation: Algorithms and Applications*. PhD thesis, University of Pennsylvania.

Rigouste L., Cappé O. and Yvon F. (2006). Quelques observations sur le Modèle LDA. In *Actes des Journées Internationales d'Analyse statistique des données textuelles*, Besançon, France, pp. 819-830.

Rigouste L., Cappé O. and Yvon F. (2007). Inference and evaluation of the multinomial mixture model for text clustering. *Information Processing and Management*, 43 (5) : 1260-1280.

Utiyama M. and Isahara H. (2001). A statistical model for domain-independent text segmentation. In *Proceedings of the annual Meeting of the Association for Computational Linguistics*, Toulouse, France, pp. 491-498.