Apprentissage d'un espace de concepts de mots pour une nouvelle représentation des données textuelles

Young-Min Kim, Jean-François Pessiot, Massih-Reza Amini, Patrick Gallinari

Laboratoire d'Informatique de Paris 6
104, Avenue du Président Kennedy
75016 Paris, France

{kim, pessiot, amini, gallinari}@poleia.lip6.fr
http://www-connex.lip6.fr

RÉSUMÉ. *Dans cet article nous proposons une technique à base d'apprentissage non-supervisé pour la réduction de dimension des données textuelles. Cette technique est basée sur l'hypothèse que les termes co-occurrants dans les mêmes documents avec les mêmes fréquences sont sémantiquement proches. Suivant cette hypothèse les termes sont d'abord regroupés avec l'algorithme CEM qui est une version classifiante de l'algorithme EM. Les documents sont ensuite représentés dans l'espace de ces groupes de termes. Nous jugeons de la pertinence de cette technique de réduction dimensionnelle avec la tâche du clustering de documents. Et nous montrons la validité de notre approche en comparant le résultat de ce clustering avec ceux obtenus dans l'espace sac-de-mots initial et l'espace des groupes de mots induit par l'algorithme PLSA sur deux collections standard de* WebKB *et de* Reuters.

ABSTRACT. *We present in this paper an unsupervised learning method for dimensionality reduction of text data. This technique is based on the hypothesis that terms co-occuring in the same context with the same frequency are semantically related. On the basis of this hypothesis we first find term clusters using a classifiant version of the EM algorithm. Documents are then represented in the space of these term clusters. We evaluate this method on the task of document clustering and show the effectiveness of our approach on two standard classification collections of* WebKB *and* Reuters.

MOTS-CLÉS : *Apprentissage non-supervisé, Partition de mots, Partitionnement de documents*

KEYWORDS: *Unsupervised learning, Term clustering, Document Clustering.*

1. Introduction

La tâche du clustering de documents est un problème important en Recherche d'Information. Les premiers travaux dans ce sens ont montré que les performances des moteurs de recherche pouvaient être améliorées en regroupant d'abord les documents suivant les différentes partitions et en appariant ensuite les requêtes sur les documents des partitions les plus similaires avec ces dernières (Van Rijsbergen, 1979). Depuis la fin des années 90, cette tâche a été massivement utilisée pour la navigation sur le Web (Cutting *et al.,* 1992), la recherche distribuée (Xu *et al.,* 1999) et le résumé automatique de textes (Kummamuru *et al.,* 2004).

La plupart des méthodes de clustering de documents reposent sur la représentation vectorielle sac de mots (Van Rijsbergen, 1979). En utilisant chaque mot comme une caractéristique, chaque document est représenté comme un vecteur de fréquences de mots (éventuellement normalisées). Avec cette approche, les documents sont représentés par des vecteurs de dimension égale à la taille du vocabulaire, qui est en général assez grand. En effet, même des collections de documents de taille moyenne peuvent contenir de nombreux mots différents, et des vocabulaires de plusieurs dizaines de milliers de mots sont désormais communs. Or la grande dimension de ces données rend la plupart des algorithmes de clustering difficiles à utiliser. À cette difficulté algorithmique vient s'ajouter le fait que les représentations des données textuelles sont typiquement creuses (Dhillon *et al.,* 2001). En effet, la plupart des documents contiennent très peu de mots par rapport à la taille du vocabulaire de la collection (typiquement moins de 5%). Il y a également le problème du bruit : les textes issus de pages web, de forums de discussion ou d'emails contiennent souvent des fautes d'orthographe et des abbréviations qui peuvent être considérés comme du bruit par rapport au texte initial. Or la plupart des algorithmes de clustering ne sont pas adaptés pour traiter de telles données. Enfin, les approches de type sac de mots ne peuvent extraire que des caractéristiques de bas niveau, sémantiquement pauvres. Il y a un fossé sémantique important avec des caractéristiques de haut niveau comme les thématiques que nous souhaitons identifier dans la collection.

Ces inconvénients sont inhérents au choix de la représentation des documents dans l'espace des mots, et ils ont motivé l'utilisation de la réduction dimensionnelle pour déterminer une nouvelle représentation plus compacte et pertinente des documents. Dans le cadre supervisé, il existe plusieurs approches pour réduire la dimension des données textuelles. Par exemple, la sélection de caractéristiques permet de réduire considérablement la dimension sans dégrader l'erreur de classification, voire même en l'améliorant dans certains cas (Yang *et al.,* 1997). Dans le cadre non supervisé en revanche, l'information de classe n'est pas disponible et la réduction dimensionnelle doit alors s'appuyer sur une connaissance *a priori* du problème. Par exemple, la nouvelle représentation extraite par l'algorithme Indexation Sémantique Latente (LSI) correspond aux axes principaux déterminés par l'analyse en composantes principales (Deerwester *et al.,* 1990). Il existe également des heuristiques simples qui permettent d'éliminer des mots jugés non informatifs, reposant notamment sur leurs fréquences

dans les documents (Salton *et al.,* 1986). Ces méthodes restent moins efficaces que des approches supervisées comme la sélection de variables, mais peuvent néanmoins réduire le bruit associé à la représentation dans l'espace des mots.

Dans cet article nous considérons une approche générale de réduction dimensionnelle non supervisée pour le clustering de documents, qui transforme l'espace des mots initial en un espace de concepts (Caillet *et al.,* 2004) . L'information *a priori* permettant cette transformation repose sur l'hypothèse $\mathcal{H}$ que *des mots qui co-occurrent avec les mêmes fréquences dans les mêmes documents sont sémantiquement proches*. Sur la base de cette hypothèse, les mots sont d'abord regroupés en clusters, appelés concepts. Puis les documents sont représentés dans le nouvel espace induit par ces concepts, où chaque nouvelle caractéristique correspond à un cluster de mots et représente le nombre total d'occurrences des mots du cluster présents dans le document. La contribution principale de ce travail est la validation empirique de notre hypothèse pour trouver des concepts de mots pertinents via la tâche du clustering. Nous utilisons le cadre du clustering de documents pour évaluer l'espace de concepts induit par notre hypothèse, et nous comparons ses performances avec l'espace de concepts induit par PLSA (Hofmann, 1999) ainsi que l'espace des sac de mots initial. Les résultats obtenus sur deux collections standard WebKB et Reuters montrent la capacité de notre approche à déterminer des concepts de mots pertinents pour la tâche, et à améliorer les performances en clustering de documents. Nous proposons ensuite une extension du modèle de PLSA pour tenir compte parallèlement des clusters de documents et de mots.

Le reste de cet article est organisé de la façon suivante. Dans la section 2 nous présentons les diffèrents modèles probabilistes pour la réduction dimensionnelle et le clustering de documents que nous avons utilisés dans cet article. Nous donnons les résultats expérimentaux dans la section 3 et dans la section 4 nous présentons une conclusion à ce travail.

2. Modèles

Nous commençons notre présentation par le modèle PLSA (section 2.2), introduit par (Hofmann, 1999), pour la recherche de concepts latents dans une collection de documents. Nous présentons ensuite nos deux contributions pour le clustering de documents. Dans la section 2.3, nous présentons l'hypothèse qui nous sert à déterminer des concepts de mots dans le cadre de la réduction dimensionnelle des données textuelles. Dans la section 2.4, nous proposons une extension du modèle PLSA, capable de dissocier les clusters de documents des thématiques trouvées dans la collection.

2.1. *Notations*

Nous notons par $\mathcal{V} = \{w_j\}_{j\in\{1,...,|\mathcal{V}|\}}$ l'ensemble des mots du vocabulaire d'une collection $\mathcal{D} = \{d_i\}_{i\in\{1,...,n\}}$ de n documents. Nous représentons les mots par

$$\vec{w} =< n(d_i, w) >_{i\in\{1,...,n\}} \quad [1]$$

où $\vec{w}$ est la représentation vectorielle du mot w, et $n(d_i, w)$ est le nombre d'occurrences du mot w dans le document $d_i \in \mathcal{D}$. Nous représentons les concepts latents par l'ensemble $A = \{\alpha_1, ..., \alpha_L\}$ comprenant L composantes.

2.2. *Probabilistic Latent Semantic Analysis (PLSA)*

Le modèle PLSA est un modèle probabiliste qui caractérise chaque mot dans un document comme une variable aléatoire générée par un modèle de mélange dont les composantes du mélange sont des distributions multinomiales. Ce modèle associe une variable latente non-observée (appelée topic ou composante) $\alpha \in A = \{\alpha_1, ..., \alpha_L\}$ à chaque observation correspondant à l'occurrence d'un mot $w \in \mathcal{V}$ dans un document $d \in \mathcal{D}$. Dans ce cas le processus de génération des mots d'une collection donnée suit le schéma graphique suivant (figure 1 (a)) :

– Choisir un document d avec une probabilité $p(d)$,

– Choisir une variable latente α d'après sa probabilité conditionnelle $p(\alpha \mid d)$,

– Génerer un mot w suivant la probabilité $p(w \mid \alpha)$.

La génération d'un mot w dans un document d peut alors être traduite par le modèle de probabilité jointe suivant :

$$p(w, d) = p(d) \sum_{\alpha\in A} p(w \mid \alpha)p(\alpha \mid d) \quad [2]$$

2.2.1. *Apprentissage du modèle*

Les paramètres du modèle [2] sont estimés suivant le principe du maximum de vraisemblance en optimisant la fonction :

$$\mathcal{L} = \sum_{d\in\mathcal{D}} \sum_{w\in\mathcal{V}} n(d, w) \log p(d, w) \quad [3]$$

La variable thématique α n'étant pas observée, les paramètres du modèle sont estimés suivant la procédure Espérance Maximisation (EM) (Dempster *et al.,* 1977). L'étape E, consiste à estimer les probabilités *a posteriori* de la variable latente α. Et, à l'étape M, on ré-estime de nouveaux paramètres du modèle en maximisant la fonction de log-vraisemblance [3].

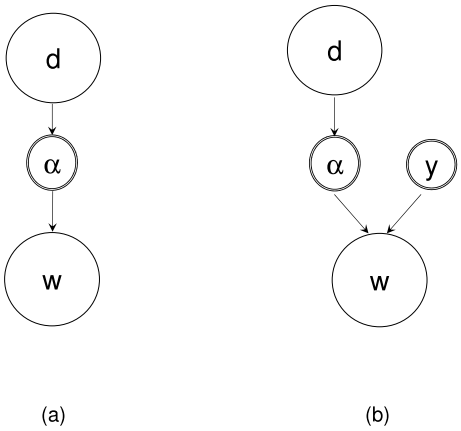

Figure 1. *Deux modèles graphiques pour le clustering de documents (a) le modèle PLSA (b) notre extension de PLSA*

2.2.2. *Classification non supervisée avec PLSA*

Avec le modèle PLSA, il est également possible de partitionner les documents d'une collection $\mathcal{D}$ donnée. En effet, la quantité $p(\alpha \mid d)$ est la probabilité d'observer la thématique α sachant le document d, et s'interprète naturellement comme la probabilité pour le document d d'appartenir à la thématique α. Ainsi, l'algorithme PLSA peut être utilisé comme un algorithme de clustering de documents, où chaque cluster correspond à une thématique. Pour partitionner les documents en clusters, il suffit alors d'attribuer à chaque document la thématique la plus probable :

$$\text{cluster}(d) = \underset{\alpha \in A}{argmax}\, p(\alpha \mid d) \qquad [4]$$

où les $p(\alpha|d)$ sont connues puisqu'elles sont apprises par le modèle. De la même manière, il est possible d'utiliser PLSA pour le clustering de mots. Pour cela, il suffit d'interpréter $p(\alpha|w)$ comme la probabilité d'appartenance du mot w à la thématique α. Les probabilités $p(\alpha \mid w)$ ne sont pas disponibles directement, mais peuvent être exprimées en fonction des paramètres du modèle. Après calculs, nous obtenons :

$$\text{cluster}(w) = \underset{\alpha \in A}{argmax}\, p(\alpha|w) = \underset{\alpha \in A}{argmax}\, p(w|\alpha) \sum_{d} p(d)p(\alpha|d) \qquad [5]$$

Nous remarquons que les variables latentes A servent dans le modèle PLSA aussi bien à partitionner les documents qu'à trouver les concepts de mots.

2.3. *Apprentissage de concepts de mots avec l'hypothèse $\mathcal{H}$*

Nous proposons de trouver les concepts de mots grâce à l'hypothèse $\mathcal{H}$ qui suppose que deux mots sont sémantiquement proches s'ils co-occurrent avec les mêmes fréquences dans les mêmes documents. Nous regroupons ainsi les mots d'une collection en différents groupes ou *concepts* et représentons les documents dans l'espace de ces concepts. Dans cette section nous présentons l'algorithme qui nous sert à trouver les concepts de mots et ensuite proposons une manière simple de représenter les documents dans l'espace des concepts ainsi trouvés.

2.3.1. *Algorithme CEM*

Pour trouver les concepts de mots nous utilisons l'algorithme Classification-Espérance-Maximisation (CEM) (Celeux *et al.,* 1992), qui est une version classifiante de l'algorithme EM (Dempster *et al.,* 1977).

Le nombre de clusters de mots K étant choisi et fixé, le partitionnement des mots w du vocabulaire se base sur un modèle probabiliste génératif des mots $\vec{w}$ et l'hypothèse simplificatrice que deux mots différents $\vec{w}$ et $\vec{w'}$ sont générés indépendamment par ce modèle génératif.

Plus formellement, nous supposons que chaque terme w est généré par le modèle de mélanges

$$p(\vec{w}|\Theta) = \sum_{k=1}^{K} \pi_k p(\vec{w}|c = k, \theta_k) \quad [6]$$

où θ_k est l'ensemble de paramètres (appris par l'algorithme) associé au cluster k, Θ est l'ensemble de tous les paramètres du modèle et $\pi_k = p(c = k|\Theta)$, la probabilité qu'un mot généré aléatoirement appartienne au cluster k.

La deuxième hypothèse est que chaque terme appartient à un seul et unique cluster. Formellement, nous associons à chaque terme $w_i \in V$ un vecteur d'indicateurs de cluster $t_i = \{t_{hi}\}_h$ tel que :

$$\forall w_i \in V, \forall k, y_i = k \Leftrightarrow t_{ki} = 1 \text{ et } \forall h \neq k, t_{hi} = 0 \quad [7]$$

2.3.2. *Apprentissage de concepts de mots*

L'algorithme CEM détermine les clusters de mots C en maximisant les paramètres Θ qui maximisent la log-vraisemblance des données complètes :

$$\mathcal{L}_{CML}(C, \Theta) = \sum_{w_j \in \mathcal{V}} \sum_{k=1}^{K} t_{kj} \log p(\vec{w}_j, y = k, \Theta)$$

Algorithm 1: Algorithme `CEM`

Input :

– Une partition initiale $C^{(0)}$ est initialisée aléatoirement et les probabilités conditionnelles de cluster $p(w \mid y = k, \theta_k^{(0)})$ sont estimées sur les clusters correspondants.

– $l \leftarrow 0$

repeat

– `Étape E` : Estimer des probabilités *a posteriori* d'appartenance du terme w_j au cluster $C_k^{(l)}$:

$$\forall w_j \in V, \forall k \in \{1, ..., K\},$$

$$\mathbb{E}[t_{kj}^{(l)} \mid \vec{w}_j; C^{(l)}, \Theta^{(l)}] = \frac{\pi_k^{(l)} p(\vec{w}_j | y=k)}{p(\vec{w}, \Theta^{(l)})}$$

– `Étape C` : Attribuer à chaque $w_j \in V$ le cluster $C_k^{(l+1)}$ de probabilité *a posteriori* maximale suivant $\mathbb{E}[t \mid w]$. Notons $C^{(l+1)}$ la nouvelle partition.

– `Étape M` : Estimer les nouveaux paramètres $\Theta^{(l+1)}$ qui maximisent :

$$\mathcal{L}_{CML}(C^{(l+1)}, \Theta^{(l)})$$

– $l \leftarrow l + 1$

until *convergence de* $\mathcal{L}_{CML}$;

Output : Concepts de mots C

Ici, les vecteurs indicateurs de clusters t font partie des paramètres du modèle et sont donc appris avec Θ. Dans nos expériences, nous avons supposé que les termes étaient générés indépendamment par le mélange de densité [6] où chaque composante du mélange $p(\vec{w}|y)$ obéit à un modèle Naïve Bayes. Les paramètres Θ du modèle sont l'ensemble des probabilités *a priori* des clusters $\pi_k = p(y = k)$ et les probabilités des documents d_i sachant les clusters $\{p_{ik}\}_{i \in \{1,...,n\}, k \in \{1,...,K\}}$. Avec ces hypothèses, la probabilité d'un mot w est $p(\vec{w} \mid y = k) = \prod_{i=1}^{n} p_{ik}^{n(d_i,w)}$

Pour estimer les paramètres π_k and p_{ik} qui maximisent la log-vraisemblance, nous dérivons $\mathcal{L}_{CML}$ et utilisons les multiplicateurs de Lagrange pour préserver les contraintes $\sum_k \pi_k = 1$ et $\forall k, \sum_{i=1}^{n} p_{ik} = 1$. Les formules de mise à jour sont :

$$\pi_k = \frac{\sum_{j=1}^{|V|} t_{kj}}{|V|}, \quad p_{ik} = \frac{\sum_{j=1}^{|V|} t_{kj} \times tf(w_j, d_i)}{\sum_{j=1}^{|V|} \sum_{i=1}^{n} t_{kj} \times tf(w_j, d_i)}$$

Une fois les concepts de mots trouvés, les documents sont alors représentés dans l'espace induit par ces concepts, où chaque nouvelle caractéristique correspond à un cluster de mots et représente le nombre total d'occurrences des mots du cluster présents dans le document.

2.4. *Extension de PLSA*

Dans cette section nous présentons notre extension du modèle PLSA décrit précédemment. Alors qu'avec PLSA les mots ne sont générés que par les thématiques, nous supposons que les mots du vocabulaire $\mathcal{V}$ sont générés conjointement par les clusters de documents et les concepts de mots. Cette supposition peut s'interpréter de la manière suivante : dans une collection il existe des thématiques de documents correspondant aux différents discours présents dans la collection. À l'intérieur de ces thématiques, les documents contiennent des sujets différents. Par exemple une thématique peut être *Sport* et les différents sujets de cette thématique sont les sujets sportifs parlant de domaines différents. Avec le modèle PLSA, on suppose que tous les documents appartenant à une thématique génèrent de la même façon les mots de cette thématique. Nous supposons que ces mots sont à la fois générés par le discours latent représenté par la thématique et les différents sujets traités dans cette thématique. Ainsi la différence principale entre notre modèle et PLSA est l'utilisation d'une variable latente supplémentaire y, qui représente les concepts de mots. Le processus génératif correspondant à notre modèle est le suivant :

– Choisir un document d suivant $p(d)$,

– Générer une thématique α d'après $p(\alpha|d)$,

– Choisir un concept de mots y suivant la probabilité $p(y)$,

– Générer un mot w d'après $p(w|\alpha, y)$.

La figure 1 (b) montre la représentation graphique de notre modèle. L'utilisation de la variable y pour modéliser les concepts de mots présente deux avantages majeurs par rapport à PLSA. Tout d'abord, grâce aux deux variables latentes α et y, le nouveau modèle est capable de capturer les thématiques sur deux niveaux sémantiques différents : α capture les thématiques générales de la collection, tandis que y capture des concepts de mots correspondant à des sous-thématiques. Ensuite, lorsque notre modèle est utilisé pour le clustering de documents, le nombre de clusters de documents (cardinalité de α) peut être choisi indépendamment du nombre de concepts de mots (cardinalité de y) présents dans la collection.

Avec notre modèle, la probabilité jointe d'observer le document d, le mot w, la thématique α et le concept de mots y est donnée par

$$p(d, w, \alpha, y) = p(d)p(\alpha \mid d)p(y \mid d, \alpha)p(w \mid d, \alpha, y)$$

Les hypothèses d'indépendance conditonnelle nous permettent ensuite d'écrire $p(y \mid d, w) = p(y)$ et $p(w|d, \alpha, y) = p(w|\alpha, y)$. Finalement la probabilité jointe se simplifie en :

$$p(d, w, \alpha, y) = p(d)p(\alpha \mid d)p(y)p(w \mid \alpha, y)$$

Ainsi la probabilité jointe d'observer le document d et le mot w est

$$p(d, w) = \sum_{\alpha \in A} \sum_{y \in \mathcal{Y}} p(d)p(\alpha|d)p(y)p(w|\alpha, y)$$

Où, $\mathcal{Y}$ est l'ensemble des concepts de mots.

2.4.1. *Apprentissage du modèle*

Nous estimons les paramètres $P(d)$, $P(\alpha|d)$, $P(y)$ and $P(w|\alpha, y)$ de notre modèle en suivant le principe du maximum de vraisemblance en optimisant la fonction de log-vraisemblance [3]. Les variables α et y n'étant pas observées, nous utilisons l'algorithme EM pour estimer les paramètres de notre modèle (Dempster *et al.,* 1977). Dans l'étape E, nous estimons les probabilités *a posteriori* des variables cachées α, y. Dans l'étape M, nous ré-estimons les paramètres du modèle qui maximisent l'espérance de la fonction [3]. Ces estimés sont :

$$p(d) = \frac{\sum_{w\in\mathcal{W}} n(d,w)}{\sum_{d'\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d',w)} \quad [8]$$

$$p(\alpha|d) = \frac{\sum_{w\in\mathcal{V}}\sum_{y\in Y} n(d,w)p(\alpha,y|d,w)}{\sum_{\alpha'\in A}\sum_{w\in\mathcal{V}}\sum_{y\in Y} n(d,w)p(\alpha',y|d,w)} \quad [9]$$

$$p(y) = \frac{\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{V}}\sum_{\alpha\in A} n(d,w)p(\alpha,y|d,w)}{\sum_{y'\in Y}\sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{V}}\sum_{\alpha\in A} n(d,w)p(\alpha,y'|d,w)} \quad [10]$$

$$p(w|\alpha, y) = \frac{\sum_{d\in\mathcal{D}} n(d,w)p(\alpha,y|d,w)}{\sum_{w'\in\mathcal{V}}\sum_{d\in\mathcal{D}} n(d,w')p(\alpha,y|d,w')} \quad [11]$$

2.4.2. *Classification non supervisée de documents*

Nous pouvons utiliser notre modèle pour le clustering de documents. Comme avec PLSA, nous interprétons la quantité $p(\alpha|d)$ comme la probabilité que le document d appartienne à la thématique α. À chaque document est donc attribué le cluster vérifiant :

$$\text{cluster}(d) = \underset{\alpha\in A}{argmax}\, p(\alpha|d) \quad [12]$$

3. Résultats Expérimentaux

Nos objectifs ici sont (a) de vérifier l'efficacité de l'espace de concepts induit par l'algorithme CEM et, (b) de montrer que la prise en compte de la variable latente associée aux clusters de mots dans la version étendue de PLSA est valide. À noter que si nos algorithmes fonctionnent de manière non supervisée, notre évaluation expérimentale est réalisée à partir de corpus de documents étiquetés. En effet, nous évaluons la pertinence de l'hypothèse $\mathcal{H}$ et celle de nos algorithmes par leur capacité à retrouver les thématiques latentes dans un corpus de documents. Nous avons donc besoin de corpus dans lesquels les documents sont déjà regroupés par thématique. Dans la suite de cette section, nous examinons d'abord les performances du clustering de documents obtenu dans l'espace de concepts induit par l'hypothèse $\mathcal{H}$, par le modèle PLSA avec le résultat du clustering de documents dans l'espace sac-de-mots. Nous avons utilisé

Tableau 1. *Les caractéristiques des collections* `Reuters` *et* `WebKB`

Reuters

Classe	size	pr. %
acq	1080	24.9
crude	295	7
earn	2002	46.2
grain	283	6.5
interest	106	2.4
money	362	8.4
trade	207	4.8

WebKB

Classe	size	pr. %
course	930	22.1
faculty	1123	26.8
project	504	12.0
student	1641	39.1

l'algorithme CEM comme technique de clustering de documents. Nous avons aussi comparé les performances de l'algortihme PLSA avec son extension introduite dans la section 2.4.

3.1. *Le corpus*

Pour l'évaluation, nous avons construit nos bases à partir de deux collections de documents standard. La première est la collection `Reuters`[1]. Nous nous sommes intéressés aux 7 classes les plus représentées (acq, crude, earn garin, interest, money, trade) dans cette collection avec un nombre total de 4335 documents. La deuxième collection est le corpus `WebKB`[2] (4 universities). Nous avons choisi de travailler sur les 4 classes les plus larges avec un nombre comprenant 4196 articles. Notre prétraitement consiste à filtrer le texte en enlevant les balises html, à convertir les majuscules en minuscules et à enlever les caractères non alpha-numériques. Nous filtrons également les mots suivant un anti-dictionnaire anglais[3] ainsi que les mots qui apparaissent dans moins de 3 documents. Le vocabulaire obtenu après ce filtrage est constitué de 6990 mots pour `Reuters` et de 11170 mots pour le corpus `WebKB`. Le tableau 1 récapitule les caractéristiques des deux collections.

Dans nos expériences, nous avons divisé chaque collection en 10 sous-ensembles en préservant dans chacun d'eux les proportions entre les différentes classes. Cette division est nécessaire pour éviter des biais causés par la structure de chaque collection et pour baisser les effets des initialisations aléatoires utilisées dans les algorithmes. Pour les résultats expérimentaux, les performances sont ainsi une moyenne de 10 performances obtenus sur chacun des sous-ensembles. Le maintien de la proportion dans chaque sous-base est important parce que nous voulons savoir si la nouvelle représentation des documents permet de mieux retrouver les petites classes que la représentation initiale dans l'espace des mots.

1. http ://www.daviddlewis.com/resources/testcollections/reuters21578/
2. http ://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/
3. http ://ir.dcs.gla.ac.uk/resources/test_collections/cacm/

3.2. *Mesure d'évaluation*

Afin d'évaluer la pertinence des partitions obtenues, nous devons savoir à quelle classe initiale correspond chaque cluster appris. Pour cela, nous suivons l'approche de (Slonim *et al.,* 2002) et nous attribuons à chaque cluster sa classe majoritaire, c'est à dire la classe originale la plus représentée parmi les documents du cluster. Et en calculant des mesures d'évaluation sur les clusters obtenus, nous pouvons comparer les performances des differentes méthodes. Afin d'évaluer les résultats, nous utilisons trois mesures d'évaluation, la micro-moyenne de précision, le rappel, et l'Information Mutuelle Normalisée (IMN).

Précision et Rappel
Pour chaque classe C_l, nous avons estimé les quantités suivantes :

– $\alpha(C_l)$: Le nombre de documents correctement affectés à C_l
– $\beta(C_l)$: Le nombre de documents incorrectement affectés à C_l
– $\gamma(C_l)$: Le nombre de documents incorrectement non affectés à C_l

La précision pour une classe est définie comme $Precision(C_l) = \frac{\alpha(C_l)}{\alpha(C_l)+\beta(C_l)}$. Le rappel pour une classe est défini comme $Rappel(C_l) = \frac{\alpha(C_l)}{\alpha(C_l)+\gamma(C_l)}$. La micro-moyenne des précisions est définie comme suit :

$$Precision = \frac{\sum_{C_l} \alpha(C_l)}{\sum_{C_l} \alpha(C_l) + \beta(C_l)}$$

D'après l'égalité d'au-dessous, nous remarquons que les micro-moyennes des précisions et des rappels sont égales. Ce résultat est aussi donné dans (Slonim *et al.,* 2002)

$$\frac{\sum_{C_l} \alpha(C_l)}{\sum_{C_l} \alpha(C_l) + \beta(C_l)} = \frac{\sum_{C_l} \alpha(C_l)}{\sum_{C_l} \alpha(C_l) + \gamma(C_l)}$$

Information Mutuelle Normalisée
L'IMN est une méthode largement utilisée pour l'évaluation du clustering. Comme (Strehl *et al.,* 2002) nous calculons et normalisons l'information mutuelle entre deux partitions, l'une correspondant à l'ensemble des vraies classes, et l'autre au partitionnement à évaluer. Comme la précision, la valeur de IMN est comprise entre 0 et 1 et elle est égale à 1 quand les deux partitions sont identiques.

Pour toutes les classes, IMN est estimé en utilisant l'équation suivante :

$$IMN = \frac{\sum_{h=1}^{c} \sum_{l=1}^{c} n_{h,l} log\left(\frac{n \cdot n_{h,l}}{n_h n_l}\right)}{\sqrt{\left(\sum_{h=1}^{c} n_h log\frac{n_h}{n}\right)\left(\sum_{l=1}^{c} n_l log\frac{n_l}{n}\right)}}$$

où n est le nombre total de documents, n_h est le nombre de documents dans le cluster C_h, n_l est le nombre de documents appartenant à la classe l, $n_{h,l}$ est l'intersection des

documents dans le cluster C_h et dans la classe l. Le c est le nombre de classes dans la collection et aussi le nombre de clusters dans nos expériences.

3.3. *Résultats*

Les expériences avec des nombres variés de concepts de mots sont nécessaires pour trouver le nombre de concepts qui donne le meilleur résultat, et pour vérifier l'influence sur les résulats du nombre de concepts. Pour trouver l'espace de concepts par CEM nous avons utilisé 10, 20, 30, 40, 50, 60, 70 concepts pour les deux collections ; nous avons également utilisé 80, 90, 100, et 150 concepts pour WebKB. Pour des valeurs plus grandes du nombre de concepts de mots, les partitions obtenues avec l'algorithme CEM comportaient des clusters vides. Nous avons donc considéré avoir atteint les nombres de concepts maximaux pour les deux collections. Pour PLSA, nous avons utilisé le même nombre de concepts que CEM. Pour le nouveau modèle, nous avons fait varier le nombre de concepts associé à y empiriquement, en considérant la taille des documents et le nombre des classes.

Tableau 2. *Mesures de précisions et de rappels (moyennées sur 10 sous-bases), micro-moyenne de précision et micro-moyenne de rappel obtenues dans l'espace de sac-de-mots et dans l'espace de concepts par l'algorithme CEM*

Reuters	Précision		Rappel	
	CEM	C-CEM	CEM	C-CEM
acq	0.43	0.65	0.60	0.77
crude	0.34	0.46	0.57	0.48
earn	0.77	0.89	0.93	0.84
grain	0.43	0.52	0.13	0.54
interest	0	0	0	0
money	0.35	0.41	0.26	0.48
trade	0	0.37	0	0.22
moyen	0.61	0.70	0.61	0.70

WebKB	Précision		Rappel	
	CEM	C-CEM	CEM	C-CEM
course	0.58	0.86	0.36	0.77
faculty	0.41	0.55	0.21	0.64
project	0.37	0.45	0.10	0.29
student	0.47	0.72	0.83	0.77
moyen	0.48	0.68	0.48	0.68

Le tableau 2 montre les performances en clustering de l'algorithme CEM lorsque les documents sont représentés dans l'espace sac-de-mots (algorithme CEM) et dans l'espace de concepts (algorithme C-CEM) sur les deux bases Reuters et WebKB. Sur les deux collections et dans la majorité des cas, les précisions, rappels et micro-moyennes de Précision de l'algorithme C-CEM sont supérieures à l'algorithme CEM. De plus, le clustering dans l'espace des concepts permet mieux d'identifier les classes de tailles moyennes dans la collection qui sont totalement phagocytées en partitionnant les documents dans l'espace sac-de-mots. Ainsi sur la collection Reuters, les rappels de chacune des classes se sont tous améliorés sauf pour la plus

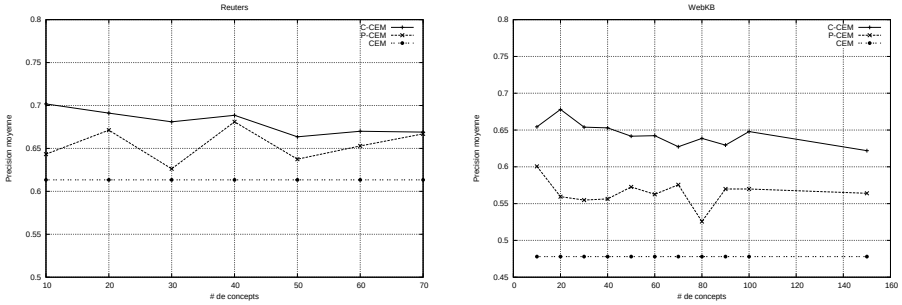

Figure 2. *Précisions moyennes de l'algorithme* `CEM` *obtenu dans l'espace sac-de-mot (*`CEM`*) et les espaces de concepts induits par l'hypothèse* $\mathcal{H}$ *(*`C-CEM`*) et par PLSA (*`P-CEM`*).*

grande classe *earn*. Alors que cette dernière a tendance à absorber les documents des autres classes lorsque nous faisons le clustering dans l'espace des mots, nous constatons que ce phénomène est atténué lorsque le clustering est effectué dans l'espace des concepts. Par exemple, la classe *trade*, qui n'a jamais été trouvée avant la réduction dimensionnelle, devient visible dans l'espace des concepts. Nous constatons des résultats similaires sur la collection `WebKB`. Après la réduction dimensionnelle en utilisant les concepts, les trois classes *course*, *faculty* et *project* sont mieux extraites et dans ce cas l'amélioration est plus grande que dans le cas de `Reuters`.

La figure 2 présente les résultats du clustering de documents sur les bases `Reuters` et `WebKB` avec l'algorithme CEM dans l'espace sac-de-mots (`CEM`) et les espaces de concepts induits avec l'hypothèse $\mathcal{H}$ (`C-CEM`) et l'algorithme PLSA (`P-CEM`) pour différents nombres de concepts de mots. Nous remarquons que les performances de l'algorithme dans l'espace induit avec l'hypothèse $\mathcal{H}$ sont généralement bien supérieures à celles obtenues dans l'espace de concepts induits par PLSA. Bien que l'espace de concepts induit par PLSA obtienne de meilleurs résultats que l'espace original, il est toujours moins bon que l'espace de concepts induit par notre hypothèse. Ces résultats montrent que la prise en compte des dépendances locales de termes via ici l'hypothèse $\mathcal{H}$ permet de trouver d'une manière pertinente les thématiques présentes dans une collection. Nous remarquons aussi que le clustering de documents dans l'espace de concepts induit par PLSA est moins bon que le clustering de documents avec l'algorithme PLSA dans l'espace des mots. Ceci nous amène à penser que si PLSA était capable de modéliser les concepts de mots d'une manière indépendante de sa modélisation des clusters de documents, ses performances en clustering de documents dans l'espace des mots pourraient être améliorées. Ces résultats rejoignent les remarques formulées à la section 2.4.

La figure 3 montre les courbes de performances des modèles PLSA et sa variante (section 2.4). Nous rappelons que l'algorithme PLSA n'utilise pas de concepts de

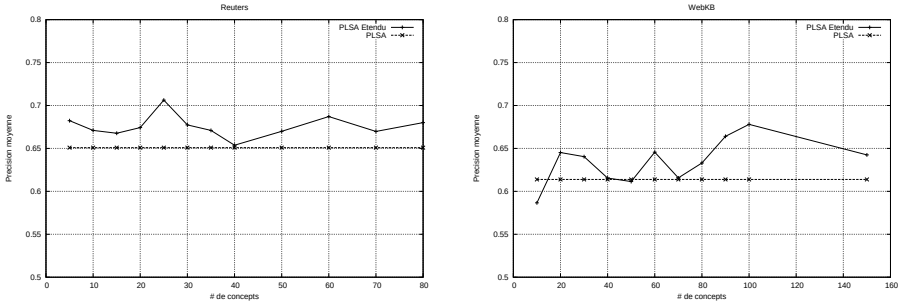

Figure 3. *Performances du clustering de documents avec PLSA et PLSA étendu*

mots, et ses résultats sont donc représentés par une ligne horizontale sur les figures. Nous remarquons de plus que sur la base `Reuters`, le modèle PLSA étendu a de meilleures performances que PLSA, ceci quel que soit le nombre de concepts choisis. Si on prend le meilleur résultat du modèle PLSA étendu, l'écart des performances entre ce modèle et le modèle PLSA est approximativement de 6%. Par manque de place, nous n'avons pas montré les performances en IMN des modèles mais elles sont semblables à celle des précisions moyennes montrées ici. Pour la base `WebKB`, les performances du modèle PLSA étendu sont presque toujours meilleures que celles de PLSA (sauf pour un nombre de concepts égal à 10). Ces résultats suggèrent que la réduction de dimension est dépendante de la dimension de départ : 20 concepts de mots sont nécessaires pour obtenir de bonnes performances sur la base `WebKB` (pour un vocabulaire initial de 11170 mots), alors que 5 concepts suffisent sur `Reuters` (pour un vocabulaire de 6990 mots). Cette constatation empirique coïncide avec l'intuition que dans un texte le nombre de mots a tendance à augmenter avec le nombre de thématiques qui sont abordées.

Nous présentons finalement au tableau 3, une comparaison entre les différentes méthodes de clustering. Les résultats présentés dans le premier tableau sont les précisions moyennes sur les 10 sous-ensembles des collections `Reuters` et `WebKB`. Nous avons calculé les précisions moyennes pour différentes valeurs de nombre de concepts de mots. La meilleure précision moyenne est la meilleure de ces moyennes. Le deuxième tableau répresente les valeurs IMN moyennes ; celles ci sont calculées pour le nombre de concepts de mots qui correspond à la meilleure précision moyenne. Comme le modèle PLSA n'utilise pas de concepts de mots, nous prenons seulement la précision moyenne et l'IMN moyen sur 10 sous-ensembles pour ce modèle. Pour les meilleures précisions moyennes, les approches `C-CEM` et PLSA étendu ont obtenu des performances similaires, elles-mêmes étant supérieures aux autres méthodes dans les deux collections. Les améliorations des deux meilleures méthodes par rapport aux autres dans la collection `WebKB` sont plus grandes que celles dans la collection `Reuters`. Pour les IMNs, les deux méthodes `C-CEM` et PLSA étendu obtiennent toujours de meilleures performances. Plus précisément la méthode `C-CEM` est légèrement

Meilleure Précision moyenne					
Collection	CEM	PLSA	P-CEM	CEM	PLSA étendu
Reuters	0.61	0.64	0.68	0.70	**0.70**
WebKB	0.47	0.61	0.60	**0.68**	**0.68**
NMI moyen correspondant à la meilleure précision moyenne					
Collection	CEM	PLSA	P-CEM	C-CEM	PLSA étendu
Reuters	0.27	0.38	0.40	**0.44**	0.42
WebKB	0.11	0.28	0.28	0.32	**0.35**

Tableau 3. *Meilleure Précision moyenne et le NMI moyen correspondant aux différents algorithmes de clustering*

meilleure que le PLSA étendu pour la collection Reuters, et inversement pour la collection WebKB. En résumé, sur les deux collections de documents et pour les deux mesures de performances, nous trouvons que les approches C-CEM et PLSA étendu ont des performances équivalentes, et sont toutes les deux meilleures que les deux algorithmes P-CEM et PLSA.

4. Conclusion

Dans cet article, nous avons proposé deux contributions au problème du clustering de documents. Notre première contribution s'incrit dans le cadre de la réduction dimensionnelle des documents, via l'utilisation des concepts de mots. Nous avons validé expérimentalement l'hypothèse selon laquelle deux mots qui co-occurrent avec les mêmes fréquences dans les mêmes documents sont sémantiquement liés, et devraient appartenir au même concept. Les expériences menées montrent que les concepts de mots obtenus permettent de déterminer une nouvelle représentation plus pertinente des documents, et d'améliorer les performances en clustering de documents. Notre seconde contribution est une extension de l'algorithme PLSA. Contrairement à l'algorithme PLSA initial, notre approche est capable de dissocier les thématiques des clusters de documents grâce à l'incorporation d'une variable modélisant les concepts de mots. Là aussi, nous avons observé une amélioration des performances en clustering de documents. Nos deux contributions confirment l'intérêt d'utiliser des concepts de mots pertinents pour traiter des données textuelles. Dans nos travaux futurs, nous voulons compléter la validation expérimentale de nos deux approches sur d'autres collections de textes plus difficiles, et également pour d'autres tâches (comme la classification supervisée par exemple). Nous voulons également étudier plus précisément les limites de notre hypothèse $\mathcal{H}$. Le fait par exemple, que deux mots synonymes ayant des occurrences différentes ne seront pas forcément regroupés dans le même concept. Une piste à explorer concerne l'utilisation de ressources linguistiques externes (comme par exemple des dictionnaires de synonymes), et plus précisément l'incorporation de ces

ressources à nos algorithmes pour aider à déterminer des concepts de mots plus pertinents.

REMERCIEMENTS

Ce travail a été partiellement financé par le programme IST de la Communauté Européenne, dans le cadre du réseau d'excellence PASCAL, IST-2002-506778 ainsi que par le projet ANR Sϵptia.

5. Bibliographie

Caillet M., Pessiot J.-F., Amini M., Gallinari P., « Unsupervised Learning with Term Clustering for Thematic Segmentation of Texts », *Proceedings of RIAO'04*, p. 648-656, 2004.

Celeux G., Govaert G., « A Classification EM algorithm for clustering and two stochastic versions », *Computational Statistics and Data Analysis*, vol. 14, n° 3, p. 315-332, 1992.

Cutting D., Karger D., Pederson J., Tukey J., « Scatter/Gatter : A Cluster Approach to Browsing Large Document Collections », *ACM SIGIR 92*, p. 318-329, 1992.

Deerwester S. C., Dumais S. T., Landauer T. K., Furnas G. W., Harshman R. A., « Indexing by Latent Semantic Analysis », *Journal of the American Society of Information Science*, vol. 41, n° 6, p. 391-407, 1990.

Dempster A. P., Laird N. M., Rubin D. B., « Maximum Likelihood from Incomplete Data via the EM Algorithm », *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, n° 1, p. 1-38, 1977.

Dhillon I. S., Modha D. S., « Concept Decompositions for Large Sparse Text Data Using Clustering », *Machine Learning*, vol. 42, n° 1/2, p. 143-175, 2001.

Hofmann T., « Probabilistic latent semantic indexing », *ACM SIGIR 99*, p. 254-261, 1999.

Kummamuru K., Lotlikar R., Roy A., Signal K., Krishnapuram R., « Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results », *In ACM WWW's04*, 2004.

Salton G., McGill M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.

Slonim N., Tishby N., « Unsupervised Document Classification using Sequential Information Maximization », *ACM SIGIR*, p. 129-136, 2002.

Strehl A., Ghosh J., « Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions », *Journal on Machine Learning Research (JMLR)*, vol. 3, p. 583-617, 2002.

Van Rijsbergen K., *Information Retrieval*, Butterworths, London, 1979.

Xu J., Croft W., « Cluster-based Language Models for Distributed Retrieval », *ACM SIGIR 99*, p. 254-261, 1999.

Yang Y., Pedersen J. O., « A comparative study on feature selection in text categorization », *in* , D. H. Fisher (ed.), *Proceedings of ICML-97, 14th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, US, Nashville, US, p. 412-420, 1997.