Cortex: un algorithme pour la condensation automatique de textes

Juan-Manuel Torres-Moreno[*,,***] — Patricia Velázquez-Morales[**] — Jean-Guy Meunier[***]**

[**] *École Polytechnique de Montréal*
Département de génie informatique C.P. 6079, Succ. Centre-ville - H3C3A7
Montréal (Québec) Canada

juan-manuel.torres@polymtl.ca

[**] *ERMETIS, Université du Québec*
555 Boul. de l'Université - G7H2B1
Chicoutimi (Québec) Canada

[***] *LANCI, Université du Québec*
C.P. 8888, Succ. Centre-Ville - H3C3P8
Montréal (Québec) Canada

RÉSUMÉ. *Étant donné que l'information sous forme électronique est déjà un standard, et que la variété et la quantité de l'information deviennent de plus en plus grandes, des méthodes d'obtention de résumés ou condensation automatique de textes constituent une phase critique de l'analyse de textes. Cet article décrit* Cortex, *un système basé sur des méthodes numériques qui permet l'obtention d'une condensation d'un texte, qui est indépendant du thème, de l'ampleur du texte et de la façon dont il est écrit. La structure du système lui permet de trouver la condensation de textes en français ou espagnol dans des temps très courts.*

ABSTRACT. *Since information in electronic form is already a standard, and that the variety and the quantity of information become increasingly large, the methods of summarizing or automatic condensation of texts is a critical phase of the analysis of texts. This article describes* Cortex *a system based on numerical methods, which allows obtaining a text condensation, which is independent of the topic and of the length of the text. The structure of the system enables it to find the abstracts in French or Spanish in very short times.*

MOTS-CLÉS : *Condensation de textes, résumés automatiques, analyse de textes, catégorisation, classification, méthodes statistiques.*

KEYWORDS: *Condensing text, automatic summarizing, texts analysis, categorization, classification, statistical methods.*

1. Introduction

La forme la plus connue et la plus visible des condensés de textes est le résumé : représentation abrégée et exacte du contenu d'un document [ANS 79]. On reconnaît trois types de résumés[MOR 99] : l'indicatif, l'informatif et le critique. Le résumé indicatif décrit le contenu du texte et aide le lecteur à décider s'il doit consulter le document original ou pas. Le résumé informatif cherche à condenser le texte de façon à ce que le lecteur n'ait pas besoin d'aller consulter le document original. Finalement, le résumé critique évalue un texte et exprime un point de vue sur le contenu. Étant donné que très peu d'applications ont besoin de résumés critiques, et que l'état de l'art ne permet pas d'obtenir d'autres types de résumés que ceux de l'indicatif ou de l'informatif, les recherches sur la génération automatique des résumés ont été portées sur la forme informative. À l'heure actuelle l'information de type électronique s'accumule rapidement et en très grande quantité sous la forme de documents qui ne sont souvent catégorisés que d'une façon très sommaire. Le manque de standards contribue à accentuer cette problématique. Les tâches de dépistage et d'exploration de l'information véhiculée dans les textes sont devenues extrêmement ardues[BUC 94]. Les méthodes linguistiques ne sont pas capables, en raison de l'ampleur et de la dynamique des corpus textuels, de faire l'analyse, la synthèse et l'extraction des connaissances dans des temps raisonnables ou faisables avec des ressources restreintes. D'un autre côté, des méthodes statistique-neuronales sont actuellement utilisées dans plusieurs domaines du traitement de l'information textuelle : dans l'indexation [SAL 71, SAL 83, DEE 90], la recherche [LEL 97], la génération des hyper-liens [VER 91], la catégorisation textuelle [BAL 96] et la classification [TOR 00, MEM 98, MEU 97, MEM 00b]. La présente recherche porte sur l'analyse statistique-numérique des textes en vue d'obtenir leur condensation informative. Notre approche a été de développer Cortex (COndensation et Résumés de TEXtes), une chaîne de traitement numérique qui inclut au cœur de sa démarche des traitements statistiques et informationnels comme des calculs d'entropie, le poids fréquentiel des segments et des mots, et plusieurs mesures d'Hamming parmi d'autres. Au-dessus des métriques se trouve un algorithme optimal de décision basé sur le vote.

2. Pré-traitement : le filtrage, la segmentation et la lemmatisation

Dans l'approche vectorielle de textes [SAL 83], on traite de documents dans leur ensemble en passant par une représentation numérique très différente d'une analyse structurale symbolique, mais qui permet des traitements performants. L'idée consiste à représenter les textes dans un espace approprié et à les appliquer des traitements vectoriels [MEM 00a]. Dans le cadre du projet Conterm (LANCI-UQAM) a été développée une chaîne de traitement numérique de textes[SEF 96]. Elle comporte des processus de filtrage, de segmentation, et de lemmatisation. Ces processus sont très performants et peuvent être appliqués à de gros corpus, car ils sont assez rapides. Le texte peut contenir des mots ou des phrases mal écrits ou incomplets, mais cette méthode tolère une certaine quantité d'erreurs aux entrées. Conterm a été large-

ment utilisé pour la classification et la catégorisation de textes en utilisant des réseaux de neurones non supervisés (ART [CAR 91], Classphères [TOR 00]) ou des méthodes statistiques ($k-$plus proches voisins ou les chaînes Markov [MEU 99]). Nous avons modifié Conterm pour l'adapter au processus spécifique des résumés automatiques. Nous commençons par un pré-traitement de données. Le texte original comporte N_M mots qui peuvent être des mots fonctionnels (articles, prépositions, adjectifs, adverbes,...), de noms ou de verbes fléchis, mais aussi des mots composés qui représentent un concept bien spécifique. Ils peuvent être répétés. On emploie plutôt la notion de *terme* pour designer un "mot" plus abstrait [MEM 00a]. Pour réduire la complexité, des processus de réduction de filtrage du lexique sont amorcés : la suppression des mots fonctionnels, des mots à haute et très basse fréquence d'apparition, la suppression du texte entre parenthèses (information additionnelle mais pas essentielle), de chiffres et des symboles spéciaux. La lemmatisation simple consiste à trouver la racine des verbes fléchis et de ramener les mots pluriels et/ou féminins au singulier masculin. Ce processus permet de diminuer la *Malédiction dimensionnelle* qui pose de très sérieux problèmes de représentation dans des grandes dimensions. La segmentation est alors faite en utilisant les *unifs* (unités d'information) : soit des termes, soit des $n-$grammes. Étant donné la nature cognitive des résumés, nous avons opté pour les termes. On va alors segmenter le texte original en petits textes (phrases), séparés selon un ou plusieurs critères adéquats. Nous avons défini comme séparateurs valables : le point, le point à la ligne, les deux points et les signes d'interrogation et d'exclamation. La longueur moyenne des segments étant fonction de la taille, si l'on désire obtenir un condensé d'un gros texte, la segmentation doit être réalisée par des paragraphes ou encore par des pages. Un indice de repérage important d'information est le titre d'un document. Toutefois nos expériences ont été réalisées sur des textes bruts, donc le titre n'est pas marqué, mais nous l'avons introduit de façon explicite comme un autre segment, ce qui permet d'augmenter la fréquence des termes qui y sont présents.

3. Condensation du texte

La segmentation transforme un document initial dans un ensemble de vecteurs où chaque segment de texte est représenté par un vecteur à composantes binaires $\vec{\Xi} = (\Xi_1, \Xi_2, \cdots, \Xi_{N_M}) = \{0,1\}^{N_M}$, où la dimension N_M est le nombre total de termes. L'ensemble des segments dont on dispose consiste en P vecteurs et la matrice $\mathbf{\Xi} = \vec{\Xi}^{\mu}; \mu = 1, \cdots, P$ représente mathématiquement le texte. Après avoir complété le prétraitement, le nouveau texte comporte un ensemble de P phrases ou segments avec N_f termes totaux, d'où on obtient un lexique de $N_{\mathcal{L}}$ termes, toujours avec la relation $N_{\mathcal{L}} \leq N_f \leq N_M$. Nous avons donc défini : $\rho_{\mathcal{L}} = \frac{N_{\mathcal{L}}}{N_M}$ le *ratio de réduction du lexique filtré/lemmatisé*. Nous utilisons seulement les termes à fréquence supérieur ou égal à 2, car nous pensons que l'information apportée par les termes à fréquence unitaire peut être négligée. On construit alors la matrice terme-segment ξ dérivée de Ξ,

à composantes $\vec{\xi} = (\xi_1, \xi_2, \cdots, \xi_{N_\mathcal{L}}) = \{0,1\}^{N_\mathcal{L}}$. La matrice $\xi = \vec{\xi^\mu}; \mu = 1, \cdots, P$ représente alors le lexique réduit du texte.

$$\xi = \begin{bmatrix} \xi_1^1 & \xi_2^1 & \xi_3^1 & \cdots & \xi_{N_\mathcal{L}}^1 \\ \xi_1^2 & \xi_2^2 & \xi_3^2 & \cdots & \xi_{N_\mathcal{L}}^2 \\ \vdots & \vdots & \vdots & & \vdots \\ \xi_1^\mu & \xi_2^\mu & \xi_3^\mu & \cdots & \xi_{N_\mathcal{L}}^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \xi_1^P & \xi_2^P & \xi_3^P & \cdots & \xi_{N_\mathcal{L}}^P \end{bmatrix} \quad [1]$$

Dans la matrice terme-segment $\vec{\xi^\mu}$, chaque composante du vecteur $\xi_i^\mu; i = 1, \cdots, N_\mathcal{L}$ montre la présence $(\xi_i^\mu = 1)$ ou l'absence $(\xi_i^\mu = 0)$ du mot i dans un segment μ. De façon analogue, la matrice fréquentielle $\mathbf{\Gamma} = \vec{\Gamma^\mu}; \mu = 1, \cdots, P$ où chaque composante $\vec{\Gamma} = (\Gamma_1, \Gamma_2, \cdots, \Gamma_{N_M})$ contient la fréquence Γ_i^μ de l'unif i dans un segment μ. Cette matrice représente l'information frequentielle complète du texte. Après le pre-traitement on obtient la matrice fréquentielle réduite γ :

$$\gamma = \begin{bmatrix} \gamma_1^1 & \gamma_2^1 & \gamma_3^1 & \cdots & \gamma_{N_\mathcal{L}}^1 \\ \gamma_1^2 & \gamma_2^2 & \gamma_3^2 & \cdots & \gamma_{N_\mathcal{L}}^2 \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_1^\mu & \gamma_2^\mu & \gamma_3^\mu & \cdots & \gamma_{N_\mathcal{L}}^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_1^P & \gamma_2^P & \gamma_3^P & \cdots & \gamma_{N_\mathcal{L}}^P \end{bmatrix} \quad [2]$$

Cette matrice représente l'information fréquentielle essentielle du texte. La condensation va s'effectuer sur l'information contenue dans les matrices réduites fréquentielle γ et binaire ξ. Elles sont creuses car chaque segment ne contient qu'une petite quantité du lexique. Nous allons définir la *taille réduite* des matrices γ et ξ comme :

$$\alpha = \frac{P}{N_\mathcal{L}} \quad [3]$$

qui représente la proportion P de segments par rapport à la dimension $N_\mathcal{L}$ du lexique réduit à l'entrée. Le texte a suivi une transformation dans des objets mathématiques, susceptibles d'être traités par des méthodes statistiques et informationnelles. Les segments ont une quantité hétérogène du lexique, qui se traduit par des vecteurs $\vec{\xi}$ plus ou moins denses. Ceci veut dire qu'il y a des segments bien plus importants que d'autres, et c'est justement cette importance que l'algorithme de condensation va identifier.

4. Algorithme

La méthode Cortex est composée de deux algorithmes : d'abord une méthode pour la construction des métriques informationnelles indépendantes sur les matrices γ et ξ,

et ensuite d'une méthode pour la récupération de l'information codée. Cette dernière fait appel à un algorithme de décision optimal, qui va agir selon une stratégie de votes qui lui seront présentés.

4.1. *Métriques*

Des informations mathématiques et statistiques importantes peuvent être dérivées à partir des matrices terme-segment ξ [1] et fréquentielle γ [2]. Nous avons regroupées ces informations dans différentes métriques, qui nous permettent de mesurer la quantité d'information contenue dans un segment. Plus un segment est importante, plus il comporte des valeurs de métriques élevés. Les 9 métriques utilisées ont été :

1. La fréquence relative des termes, 2. Les interactions de mots entre segments, 3. Les probabilités fréquentielles de mots, 4. L'entropie des segments, 5. Les distances d'Hamming entre mots, 6. Le poids d'Hamming des segments, 7. Le poids d'Hamming de mots par segment, 8. Les poids lourds d'Hamming, 9. La somme fréquentielle des poids d'Hamming.

Nous allons détailler ici seulement quelques unes des métriques utilisées dans la conception du système :

– Mesures frequentielles.

- Fréquence des mots. La somme des fréquences des unifs par segment calcule un poids spécifique d'un segment μ en utilisant l'expression :

$$F^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \gamma_i^{\mu} \qquad [4]$$

γ_i^{μ} est la fréquence du mot i dans le segment μ.

- Somme fréquentielle des probabilités Δ. Calculons d'abord les probabilités de termes. Soit p_i la probabilité d'apparition du terme i dans le texte :

$$p_i = \frac{1}{T} \sum_{\mu=1}^{P} \gamma_i^{\mu} \qquad [5]$$

où T est le nombre de termes totaux Alors la somme fréquentielle des probabilités est calculé comme :

$$\Delta = \sum_{i=1}^{N_{\mathcal{L}}} p_i \gamma_i^{\mu} \text{ ; si } \xi_i^{\mu} \neq 0 \qquad [6]$$

– Mesures entropiques. L'entropie d'un segment μ nous la calculons en utilisant :

$$E^{\mu} = -\sum_{i=1}^{N_{\mathcal{L}}} x_i^{\mu} \log_2 x_i^{\mu} \qquad [7]$$

avec $x_i^{\mu} = \gamma_i^{\mu} / \sum_{i=1}^{N_{\mathcal{L}}} \gamma_i^{\mu}$

– Mesures d'Hamming. Plusieurs mesures dérivées des distances d'Hamming ont été considérées. En étant indépendants elles ont été prises comme métriques valables. Une distance de Minskowski : $d(\vec{a},\vec{b}) \equiv \sum_i \left\|a^i - b^i\right\|$ a été utilisée comme mesure de base.

- Les distances d'Hamming Ψ. Cette quantité mesure la distance entre paires d'unifs i et j dans l'espace des segments. Chaque unif étant représentée par $\vec{\xi_i} = \{0,1\}^P$. Il faut d'abord, calculer la matrice d'Hamming H, qui est diagonale supérieure à dimension $N_{\mathcal{L}}$.

$$H_i^{i+1} = \left\{ \begin{array}{l} 1 \text{ ; si } \xi_i^\mu \neq \xi_j^\mu \\ 0 \text{ autrement} \end{array} \right\} \begin{array}{c} i=1,\cdots,N_{\mathcal{L}}-1 \\ j=i+1,\cdots,N_{\mathcal{L}} \\ \mu=1,\cdots,P \end{array} \quad [8]$$

Ensuite on calcule la somme des distances d'Hamming par segment comme :

$$\Psi^\mu = \sum_{i=1}^{N_{\mathcal{L}}} \sum_{j=i+1}^{N_{\mathcal{L}}} H_i^j \text{ si } \left(\xi_i^\mu, \xi_j^\mu\right) \neq 0 \quad [9]$$

- Le poids d'Hamming des segments. Chaque segment possède un « poids » ϕ^μ, qui est égal à la somme des unifs présentes dans le segment.

4.2. *Algorithme de décision*

Supposons qu'un ensemble de k votants indépendants ont été entraînés sur les matrices terme-segment ξ et fréquentielle γ. Chacune des métriques possède un degré de l'importance informationnelle de chaque segment. On a besoin d'un algorithme qui permet de récupérer les connaissances codées sur les votants. Ce problème peut se poser ainsi : étant donné les votes pour un événement particulier qui provient d'un ensemble de k votants indépendants (chacun avec une certaine probabilité d'avoir raison), trouver la décision optimale. La méthode que nous avons développée s'appelle *Algorithme de décision* (AD), et permet d'utiliser les connaissances partielles que possède chaque votant pour choisir typiquement les segments les plus pertinents. Tous les votants sont ajoutés un après l'autre en additionnant à chaque pas leur connaissance, c'est-à-dire la probabilité du choix de chaque segment, en augmentant ou parfois en diminuant la connaissance globale, jusqu'à la convergence ; ce qui permet à la fin de trouver l'ensemble de segments probabilistiquement les plus importants qui rendent un condensé optimal. L'AD utilise deux probabilités mutuellement exclusives : p_0 et p_1. On présente les k votants l'un après l'autre, en modifiant p_0 et p_1 à chaque occasion en fonction des sorties $\pi_j;\ j = 1,\cdots,k$ des votants ν_j sur chaque segment :

1) Pour chaque segment $\vec{\xi}^\mu$; $\mu = 1,2,\cdots,P$

2) $p_0^{(0)} \leftarrow p_1^{(0)} \leftarrow 1$; initialisation des probabilités

3) Pour $j = 1,\cdots,k$; votants ν

a) Le votant ν_j décide qu'un segment μ est pertinent avec une probabilité π_j.

b) Si la probabilité π_j est significative on modifie p_0 et p_1 à l'itération $t+1$ en fonction des valeurs à l'itération t :

$$\text{Si } \left(\pi_j \geq \frac{1}{2}\right) \quad \text{alors } \begin{cases} p_0^{(t+1)} \leftarrow p_0^{(t)}\pi_j \\ p_1^{(t+1)} \leftarrow p_1^{(t)}(1-\pi_j) \end{cases} \quad [10]$$

$$\text{sinon } \begin{cases} p_0^{(t+1)} \leftarrow p_0^{(t)}(1-\pi_j) \\ p_1^{(t+1)} \leftarrow p_1^{(t)}\pi_j \end{cases} \quad [11]$$

4) À la fin de la présentation des k votants, nous mesurons les probabilités de décision : Si $(p_0 > p_1)$ alors l'AD décide OUI au choix du segment μ. Autrement il décide NON.

Cet algorithme possède deux propriétés intéressantes : il converge, car les probabilités p_0 et p_1 sont modifiées de façon mutuellement exclusive, ce qui assure que l'écart entre p_0 et p_1 est changé avec une probabilité supérieure ou égale à $\frac{1}{2}$ de l'améliorer. Il est un amplificateur : la probabilité de choisir un segment pertinent est supérieure ou égale à la probabilité π_j du meilleur votant qu'on a branché à ce moment. Étant donné que l'AD a besoin de valeurs normalisées entre $[0, 1]$ et que certaines des métriques produisent des résultats négatifs (mesures entropiques), toutes les sorties ν_j des k votants ont été normalisées à $\hat{\nu}_j = 1 - \left(\frac{M-\nu_j}{M-m}\right)$ avec : $M = \max\{\nu_j\}$ et $m = \min\{\nu_j\}; j = 1, 2, \cdots, k$.

5. Expériences et résultats

Nous avons fait de tests sur 7 articles de vulgarisation (5 en français, 2 en espagnol). Les textes sont de petite taille (entre 140 et 1000 mots). La raison de traiter des petits textes a été de permettre la comparaison de nos condensés avec ceux obtenus par des humains. L'objectif a été, dans tous les cas d'obtenir un condensé de 25% du nombre total de segments. Nous avons fait des comparaisons avec le synthétiseur *Minds* (New Mexico State University)[1], avec *Summarizer*© (*Copernic*)[2] et avec *Word*© (*Microsoft*). Nous avons demandé à 14 personnes de lire le texte et de choisir les phrases qui leur semblaient les plus pertinentes. Tous les sujets ont un niveau universitaire et sont habitués à faire des résumés. Pour *Summarizer* nous avons comparé les résultats de [HUO 00]. Dans le cas de *Minds* et *Word* on leur a demandé d'obtenir une synthèse de 25%. Nous présenterons seulement les résultats des tests sur « Puces », texte artificiellement ambigu car composé d'un mélange appartenant à deux auteurs différents. La première partie[3] traite les puces électroniques et la deuxième[4] de la présence de puces et de poux chez les militaires. Évidemment on ne donne pas cette connaissance préalable aux systèmes. « Puces » contient $N_M = 605$

1. http://messene.nmsu.edu/minds/SummarizerDemoMain.html
2. http ://www.copernic.com
3. http ://www.admin.ch/cp/f/1997Sep10.064053.8237@idz.bfi.admin.ch.html
4. http ://www.tregouet.org/lettre/1999/Lettre47-Au.html

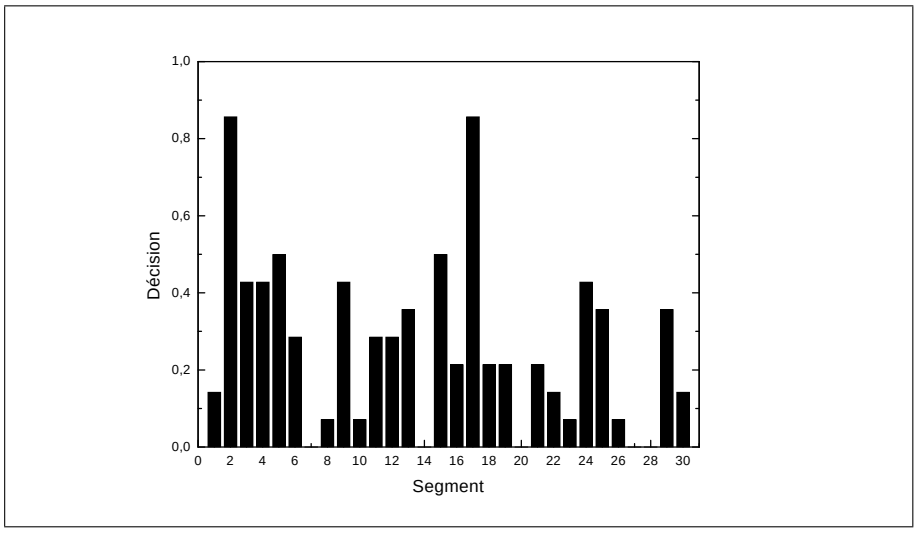

FIG. 1. *Choix de segments fait par 14 sujets. Les personnes attribuent une importance prépondérante au segment 2 (« informatique ») et au segment 17 (« biologique »).*

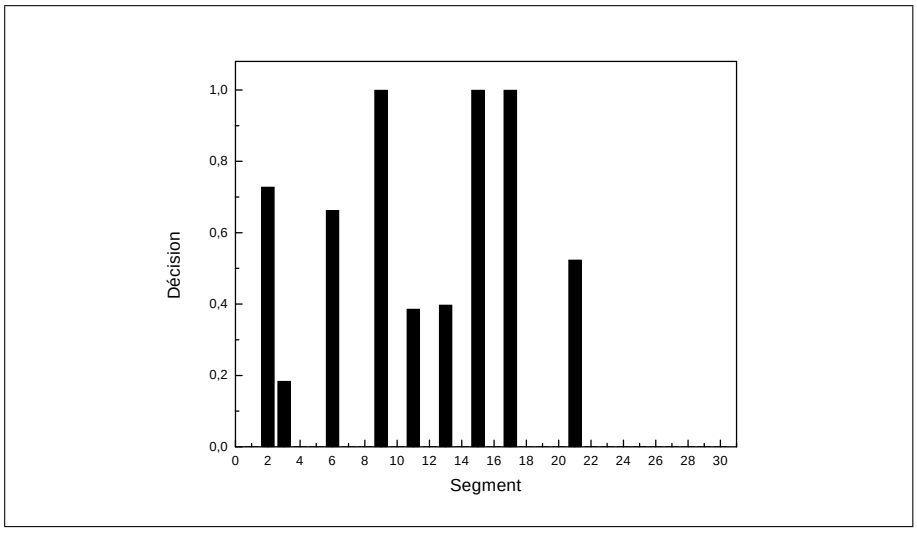

FIG. 2. *Choix de segments par* Cortex. *Les segments 2 et 17 ont été bien repérés.*

mots, $P = 30$ phrases et un lexique filtré/lemmatise $N_{\mathcal{L}} = 30$ termes. Les segments 1 à 15 traitent le thème « puces info » et le reste celui de « puces et poux ». Sur la figure 1 nous montrons les segments choisis par les humains. L'axe horizontal montre le numéro de segment et l'axe vertical la fréquence normalisé du nombre de sujets ayant choisi un segment particulier. Malgré l'écart, on constate un accord sur les ex-

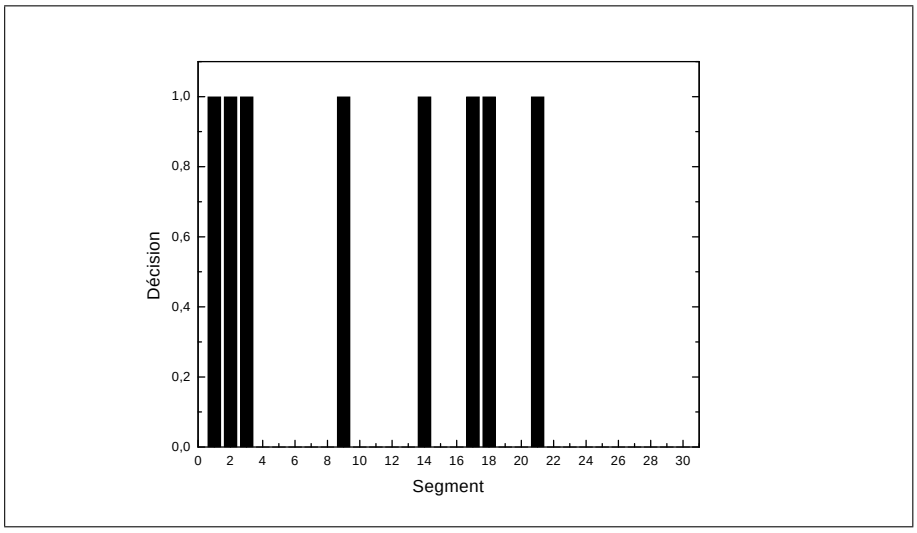

FIG. 3. *Choix de segments pertinents fait par le système* Minds*.*

trêmement importants segments 2 et 17. Dans la figure 2 nous montrons nos résultats. L'axe vertical représente la valeur fournie par l'AD. Un seuil variable en fonction de la taille voulue du condensé (25%) permet de choisir les 8 segments les plus importants. Les segments 2 et 17 ont été bien repérés. Pour les autres méthodes l'axe vertical représente un choix (oui=1, non=0) de leur décision. Sur la figure 3 nous montrons les résultats de *Minds* : on constate un résumé assez équilibré, cependant il ne trouve ni le segment 5 ni le 15 (choisis par une grande quantité de sujets). Finalement la figure 4 montre le condensé de *Word* : on voit un résumé non balancé. Bien que la partie du texte puces-info est plus important, *Word* trouve un résumé biaisé vers les segments bio. *Word* ignore les segments importants sélectionnés par les humains. Les résultats pour autres textes sont aussi très encourageants, car Cortex trouve des condenses pertinents même pour de textes à lexique faible.

6. Discussion

Le pre-traitement du texte réduit de plus en plus le lexique : $N_{\mathcal{L}} \leq N_f \leq N_M$. Des études sur les ratios de réduction du lexique ont été effectués, ce qui nous a permis d'établir un estimateur $\hat{\rho}_{\mathcal{L}}$ pour le lexique filtré/lemmatisé $\rho_{\mathcal{L}}$. La réduction de la taille du lexique filtré/lemmatisé $\hat{\rho}_{\mathcal{L}}$ suit un comportement linéaire par rapport au nombre de termes du texte original, approximativement en divisant sur seize le nombre de termes employés. Ces réductions permettent de diminuer la malédiction dimensionnelle. D'un autre côté, nos expériences ont montré que l'ordre de présentation des segments du texte n'a aucune influence sur les performances de l'algorithme. Par contre, des tests sur *Minds* et *Word* montrent qu'ils sont dépendants de cet ordre.

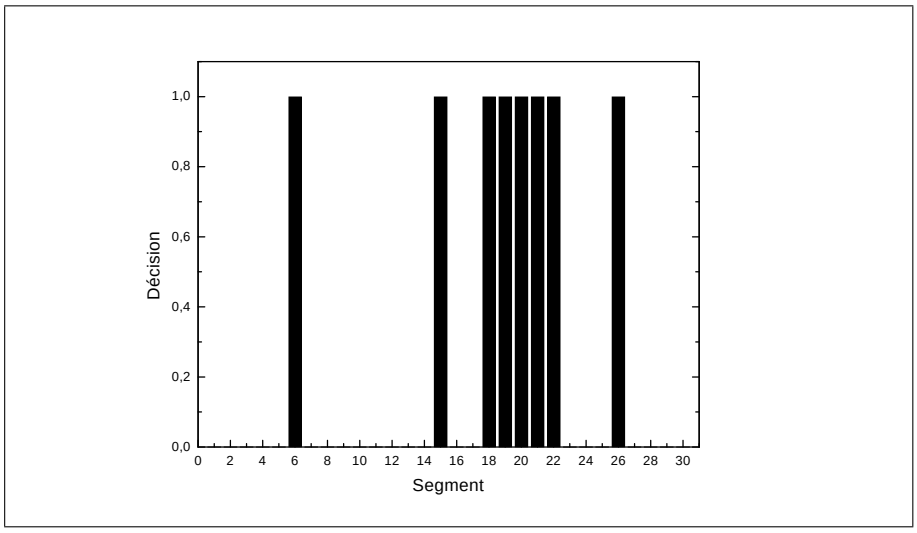

FIG. 4. *Choix de segments du synthétiseur de* Word©. *Le choix est tout à fait déséquilibré et peu pertinent.*

7. Conclusion

L'algorithme Cortex semble être un condensateur de textes très performant. Les tests faits en comparaison avec des sujets humains ou d'autres méthodes de condensation ont montré que l'algorithme est typiquement plus cohérent pour trouver les segments de texte pertinents. On obtient un résumé balancé : la plupart des thèmes sont abordés. Les avantages supplémentaires consistent à ce que les résumés sont obtenus de façon indépendante de la taille, des sujets abordés ou d'une certaine quantité de bruit. Évidemment, il s'agit d'une condensation informative, et pas d'un résumé dans le sens cognitif ; mais pour le moment toutes les recherches sont incapables d'en faire autrement. L'algorithme de décision est robuste, amplificateur des probabilités et indépendant de la présentation des segments, ce qui n'est pas le cas pour les autres méthodes. L'ajout d'autres métriques entropiques et d'un identificateur automatique de langues ne pourraient qu'améliorer la qualité des résultats. Pour terminer, Cortex est une méthode rapide et indépendante de la langue, qualités qui la rendent spécialement utile pour la condensation de grosses quantités de documents, comme sur le Web, par exemple.

8. Bibliographie

[ANS 79] ANSI, *American National Standards for Writing Abstracts*, ANSI Inc., USA, 1979.

[BAL 96] BALPE J., LELU A., PAPY F., SALEH I., *Techniques avancées pour l'hypertexte*, Éditions Hermès, Paris, 1996.

[BUC 94] BUCKLEY C., SALTON G., ALLAN J., « Automatic stucturing and retrieval of large text file », *ACM*, vol. 2, n° 37, 1994, p. 97-107, ACM.

[CAR 91] CARPENTER G., GROSSBERG S., ROSEN D., « Fuzzy ART : fast stable learning and categorization of analog patterns by an adaptive resonance system », *Neural Networks*, vol. 4, 1991, p. 759-771.

[DEE 90] DEERWESTER S., DUMAIS D., FURNAS T., LAUNDER G., HARSHMAN T., « Indexing by latent semantic analysis », *Journal of the Amer. Soc for Infor. Science*, vol. 6, n° 41, 1990, p. 391-407.

[HUO 00] HUOT F., « Copernic Summarizer ou la tentation de l'impossible », *Québec Micro*, vol. 6.12, n° 12, 2000, p. 61-64.

[LEL 97] LELOUP C., *Moteurs d'indexation et de Recherche*, Eyrolles, 1997.

[MEM 98] MEMMI D., GABI K., MEUNIER J.-G., « Dynamical Knowledge extraction from texts by ART networks », *Proc. of the NEURAP'98*, Marseille, 1998.

[MEM 00a] MEMMI D., « Le modèle vectoriel pour le traitement de documents », Cahiers Leibniz n° 2000-14, November 2000, INPG.

[MEM 00b] MEMMI D., MEUNIER J.-G., « Proc. of NC'2000 », *Using competitive networks for text mining*, Berlin, May 2000.

[MEU 97] MEUNIER J.-G., NAULT G., « Approche connexioniste au problème de l'extraction de connaissances terminologiques à partir de textes », *Les Techniques d'Intelligence Artificielle Appliquées aux Technologies de l'Information*, Lepage R. and Noumeir R. Les Cahiers scientifiques de l'ACFAS No. 90, 1997, p. 62-76.

[MEU 99] MEUNIER J.-G., REMAKI L., FOREST D., « Use of classifiers in Computer assisted reading and analysis of text (CARAT) », *Proc. of the 1999 International Conference on Imaging Science, Systems and Technology (CISST'99)*, Las Vegas, Nevada, USA, 1999.

[MOR 99] MORRIS A., KASPER G., ADAMS D., « The effects and Limitations of Automated Text Condensing on Reading Comprehension Performance », *Advances in automatic text summarization*, The MIT Press, U.S.A, 1999, p. 305-323.

[SAL 71] SALTON G., *The SMART Retrieval System - Experiments un Automatic Document Processing*, Englewood Cliffs, 1971.

[SAL 83] SALTON G., MCGILL M., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[SEF 96] SEFFAH A., MEUNIER J.-G., « Aladin : an integrated object-oriented environment for computer assited text analisys », Cahiers de recherche n° 96.1, 1996, LANCI-UQAM.

[TOR 00] TORRES-MORENO J.-M., VELAZQUEZ-MORALES P., MEUNIER J., « Classphères : un réseau incrémental pour l'apprentissage non supervisé appliqué à la classification de textes », *Actes des 5es Journées Internationales d'Analyse Statistique des Données Textuelles JADT 2000*, Lausanne, 9-11 Mars 2000, EPFL M. Rajman & J.-C. Chappelier éditeurs, p. 365-372.

[VER 91] VERONIS J., IDE N., HARIE S., « Very Large Neural Networks as a Model of Semantic Relations », *Proc. of the 3rd Cognitiva Symposium*, Madrid, 1991.