Du textuel au numérique : analyse et classification automatiques

Juan-Manuel Torres-Moreno

▶ **To cite this version:**

Juan-Manuel Torres-Moreno. Du textuel au numérique : analyse et classification automatiques. Interface homme-machine [cs.HC]. Université d'Avignon, 2007. tel-00390068

HAL Id: tel-00390068

https://theses.hal.science/tel-00390068v1

Submitted on 31 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

Diplôme d'Habilitation à Diriger des Recherches

Université d'Avignon et des Pays de Vaucluse

Spécialité : **Informatique**

Laboratoire Informatique d'Avignon (EA 931)

Du textuel au numérique : analyse et classification automatiques

Juan Manuel TORRES-MORENO

Présenté publiquement le 12 décembre 2007 devant un jury composé de :

M	Frédéric ALEXANDRE	Directeur de recherches, INRIA-Loria, Nancy	Rapporteur
M	Joan CABESTANY	Professeur, UPC, Barcelona	Rapporteur
M	Jean-Paul HATON	Professeur, INRIA-Loria, Nancy	Rapporteur
M	Guy LAPALME	Professeur, RALI, Montréal	Rapporteur
M	Eitan ALTMAN	Directeur de recherches, INRIA Sophia-Antipolis	Examinateur
M	Marc EL-BÈZE	Professeur, LIA, Avignon	Examinateur
Mme	Mirta GORDON	Directeur de recherches, CNRS-TIMC, Grenoble	Examinatrice
M	Jean-Guy MEUNIER	Professeur, LANCI, Montréal	Examinateur
Mme	Violaine PRINCE	Professeur, LIRMM, Montpellier	Examinatrice

Laboratoire Informatique d'Avignon

Résumé

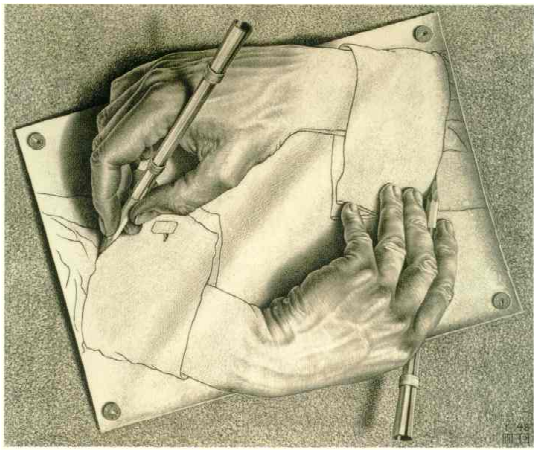
Dans ce document, je présente les travaux de recherche que j'ai menés après ma thèse, d'abord comme chercheur au LANIA, Mexique, puis pendant mon post-doctorat au Canada au LANCI-UQAM et comme chercheur au ERMETIS, ensuite à l'École Polytechnique de Montréal et finalement au LIA où je suis actuellement responsable de la thématique TALNE. Un goût personnel pour les méthodes d'apprentissage automatique m'a orienté vers leur utilisation dans le Traitement Automatique de la Langue Naturelle. Je laisserai de côte des aspects psycholinguistiques de la compréhension d'une langue humaine et je vais m'intéresser uniquement à la modélisation de son traitement comme un système à entrée-sortie. L'approche linguistique possède des limitations pour décider de cette appartenance, et en général pour faire face à trois caractéristiques des langages humaines : Ambiguïté. Je pense que l'approche linguistique n'est pas tout à fait appropriée pour traiter des problèmes qui sont liés à un phénomène sous-jacent des langues humaines : l'incertitude. L'incertitude affecte aussi les réalisations technologiques dérivées du TAL : un système de reconnaissance vocale par exemple, doit faire face à de multiples choix générés par une entrée. Les phrases étranges, mal écrites ou avec une syntaxe pauvre ne posent pas un problème insurmontable à un humain, car les personnes sont capables de choisir l'interprétation des phrases en fonction de leur utilisation courante. L'approche probabiliste fait face à l'incertitude en posant un modèle de langage comme une distribution de probabilité. Il permet de diviser un modèle de langage en plusieurs couches : morphologie, syntaxe, sémantique et ainsi de suite. Tout au long de cette dissertation, j'ai essayé de montrer que les méthodes numériques sont performantes en utilisant une approche pragmatique : les campagnes d'évaluation nationales et internationales. Et au moins, dans les campagnes à portée de ma connaissance, les performances des méthodes numériques surpassent celles des méthodes linguistiques. Au moment de traiter de grandes masses de documents, l'analyse linguistique fine est vite dépassée par la quantité de textes à traiter. On voit des articles et des études portant sur Jean aime Marie et autant sur Marie aime Jean ou encore Marie est aimée par Jean. J'ai découvert tout au long de mes travaux, en particulier ceux consacrés au résumé automatique et au raffinement de requêtes, qu'un système hybride combinant des approches numériques à la base et une analyse linguistique au sommet, donne de meilleures performances que les systèmes pris de façon isolée. Dans l'Introduction je me posais la question de savoir si la linguistique pouvait encore jouer un rôle dans le traitement de la langue naturelle. Enfin, le modèle de sac de mots est une simplification exagérée qui néglige la structure de la phrase, ce qui implique une perte

importante d'information. Je reformule alors les deux questions précédentes comme ceci : Les approches linguistiques et les méthodes numériques peuvent-elles jouer un partenariat dans les tâches du TAL ? Cela ouvre une voie intéressante aux recherches que je compte entreprendre la conception de systèmes TAL hybrides, notamment pour la génération automatique de texte et pour la compression de phrases. On peut difficilement envisager de dépasser le plafond auquel les méthodes numériques se heurtent sans faire appel à la finesse des approches linguistiques, mais sans négliger pour autant de les valider et de les tester sur des corpora [1].

[1] Résumé généré automatiquement par CORTEX 3.8 (métriques ALFX, taux de compression de 10%) à partir de quelques sources LaTeXdu manuscrit.

Table des matières

De l'écrit au numérique



M.C. Escher. Mains dessinant. Lithographie, 1948
All M.C. Escher works (c) 2007 The M.C. Escher Company - the Netherlands.
All rights reserved. Used by permission. www.mcescher.com

Dans ce document, je présente les travaux de recherche que j'ai menés après ma thèse, d'abord comme chercheur au LANIA[2] Mexique, puis pendant mon post-doctorat au Canada au LANCI-UQAM[3] et comme chercheur au ERMETIS[4], ensuite à l'École Polytechnique de Montréal[5] et finalement au LIA où je suis actuellement responsable de la thématique TALNE[6]. L'ensemble de mes recherches, tout au long de presque dix ans, a été possible grâce à la collaboration avec des collègues de trois pays différents, de mes étudiants en DEA, Master ou des chercheurs en thèse. Dans cette dissertation j'essaye de montrer une vision cohérente de mes travaux, même si la chronologie n'est pas forcément respectée. Il n'aurait pas pu en être autrement. Le double fil conducteur de mes recherches, donc du manuscrit, reste l'apprentissage automatique et le traitement de textes. Un goût personnel pour les méthodes d'apprentissage automatique m'a orienté vers leur utilisation dans le Traitement Automatique de la Langue Naturelle (TAL).

Délibérément, j'ai décidé d'utiliser un minimum de ressources ou de méthodes lin-

[2] *Laboratorio Nacional de Informática Avanzada*, http://www.lania.mx
[3] Laboratoire d'Analyse Cognitive de l'Information http://www.lanci.uqam.ca
[4] UQAC http://wwwdsa.uqac.ca/ermetis
[5] http://www.polymtl.ca
[6] http://www.lia.univ-avignon.fr/equipes/TALNE

guistiques. Mis à part la lemmatisation et quelques outils mineurs, mes recherches ne font pas appel à ces techniques. Le choix n'était pas anodin : je voulais à tout moment, créer des algorithmes à la fois indépendants de la langue et du domaine d'application. Les ressources linguistiques sont, par définition, très liées à une langue. Souvent elles le sont aussi à un domaine particulier. Le choix numérique s'imposait donc comme une issue prometteuse. À mon avis, il fallait en tirer le maximum de profit. Je vais justifier ce choix initial.

Un modèle peut être validé empiriquement de plusieurs façons. Je laisserai de côte des aspects psycholinguistiques de la compréhension d'une langue humaine et je vais m'intéresser uniquement à la modélisation de son traitement comme un système (boîte noire) à entrée-sortie. Mais comment définir l'entrée et la sortie dans le processus de la langue naturelle ? On peut le définir en plusieurs sens : de la façon statistique (corrélations des mots dans les phrases) à l'analyse profonde de la structure linguistique. Je vais prendre en exemple le traitement de la syntaxe. Elle joue un rôle central dans le TAL, car les outils subséquents (comme l'analyse sémantique par exemple) en ont besoin. Le paradigme linguistique est basé sur la notion qu'une langue peut être décrite en utilisant une grammaire formelle (comme les grammaires CFG). Une grammaire est un ensemble de règles qui spécifient comment la paire phrase-analyse est reconnue comme appartenant (grammaticalement correcte) ou pas (non-grammaticale) à une langue. L'approche linguistique possède des limitations pour décider de cette appartenance, et en général pour faire face à trois caractéristiques des langages humaines :

- Ambiguïté. Les humains sont capables de choisir l'analyse pertinente d'une phrase. En opposition, une grammaire souvent associe des multiples analyses à une phrase, sans donner d'indication pour choisir parmi elles.
- Robustesse. Les personnes sont capables de comprendre des déviations non-grammaticales. Par contre, de petites variations dans une phrase ne sont pas acceptées par une grammaire.
- Performances. Les humains sont capables de traiter les phrases complexes de façon efficace. Ces mêmes phrases posent des défis formidables aux approches linguistiques, et leurs analyses sont souvent incorrectes.

Je pense que l'approche linguistique n'est pas tout à fait appropriée pour traiter des problèmes qui sont liés à un phénomène sous-jacent des langues humaines : l'incertitude. L'ambiguïté se manifeste à plusieurs niveaux : sens (puce, le petit animal vs. puce électronique), étiquettes grammaticales (*la belle ferme le voile* : ART-ADJ | NOM-ADJ | NOM | VERB-ART | PRON-NOM | VERB), structure syntaxique (*I saw the man with the telescope* possède au moins deux structures)... L'ambiguïté fait partie des langues humaines et en fait sa richesse, mais certainement elle pose un énorme problème aux grammaires. Les grammaires formelles sur-génèrent les analyses possibles comme une fonction exponentielle du nombre de mots dans une phrase (Martin et al., 1987). L'incertitude affecte aussi les réalisations technologiques dérivées du TAL : un système de reconnaissance vocale par exemple, doit faire face à de multiples choix générés par une entrée. Il doit choisir le meilleur. Pour une personne le problème est différent. Malgré le fait que l'ambiguïté est liée à l'incertitude, une personne a accès à de vastes ressources extra-linguistiques (connaissance du monde, préférences culturelles, expériences...). L'humain est habitué à traiter la langue avec des connaissances incom-

plètes. Par contre, une grammaire n'a pas accès à ces connaissances. Elle est *a priori* non adéquate pour résoudre l'ambiguïté car elle ne peut pas faire face à l'incertitude. La robustesse est une caractéristique des langues naturelles difficile à comprendre. Les entrées non-grammaticales (par exemple : *John not home* où il manque le verbe et *l'artiste peins la nuit* où il manque l'accord au 3ème personne, mais elles restent encore compréhensibles) induisent l'utilisation non commune de la langue (*pie niche haut, oie niche bas*). Je mets mes propres phrases de ce manuscrit comme un exemple auquel la linguistique peut se heurter. Les grammaires formelles limitent les frontières de la langue et donc les phrases non-grammaticales sont classés comme non appartenant à la langue. Les phrases étranges, mal écrites ou avec une syntaxe pauvre (chat, SMS, e-mail...) ne posent pas un problème insurmontable à un humain, car les personnes sont capables de choisir l'interprétation des phrases en fonction de leur utilisation courante. Mais la robustesse sort aussi du domaine de la linguistique. La performance n'est pas uniquement un aspect technologique du TAL. La vitesse de traitement des algorithmes TAL est fonction directe de l'espace de leur représentation. L'incertitude donne lieu a de grands espaces qui sont difficiles à traiter.

L'approche probabiliste fait face à l'incertitude en posant un modèle de langage comme une distribution de probabilité. Ainsi, en cas d'ambiguïté, chaque analyse a associé une probabilité $0 \leq p \leq 1$ qui indique, le degré d'appartenance d'une phrase à la langue. Les probabilités ne sont pas uniquement des valeurs entre 0 et 1. Elles sont en accord avec les axiomes de la Théorie des probabilités, qui peut être vue comme une modélisation adéquate de la notion intuitive de la probabilité des événements (disjoints ou pas). La démarche empirique pour estimer les probabilités est l'objet de la statistique, qui est une interprétation de la théorie des probabilités. L'approche probabiliste a plusieurs avantages : i/ elle étend la théorie des ensembles de façon naturelle afin de traiter l'incertitude, avec des implications directes dans l'ambiguïté, la robustesse et la performance ; ii/ elle offre une interprétation directe et empirique à partir de la Statistique ; iii/ elle possède un lien direct avec les théories de l'apprentissage et de l'Apprenabilité et enfin iv/ il y a des avantages méthodologiques importants : tels que les techniques d'optimisation et la décomposition des modèles. Comment l'approche probabiliste traite l'incertitude ? Si l'on revient à l'exemple de l'analyseur syntaxique, il peut être posé comme un problème d'optimisation : l'analyseur ne décide pas de la grammaticalité mais il choisit l'analyse optimale qui diminue l'ambiguïté parmi toutes les possibles analyses de la phrase. Ainsi les analyses sont triées par rapport à leurs probabilités d'occurrence (sous l'hypothèse raisonnable des événements disjoints). Les probabilités sont estimées pour s'ajuster à ce qui est intuitivement acceptable. Dans ce point clé, l'utilisation des corpus est alors incontournable. L'approche probabiliste met à disposition des techniques pour pallier le problème de la robustesse. Elle permet de traiter aisément les événements de probabilité zéro (dus aux limitations des corpora, à la créativité de tout langage humain et à la tendence transgressive) avec des techniques comme le lissage (Manning et Schütze, 1999). Les méthodes probabilistes ont une interprétation directe avec la statistique. La probabilité d'un événement, notion théorique, est estimée à partir des données (les corpora) sous la supposition que l'estimateur de fréquence relative converge vers la véritable distribution de probabilité. Cependant cette question, qui peut être mal acceptée *a priori*, est formulée comme l'ap-

prentissage statistique : étant donné un ensemble fini de données, l'apprentissage automatique estime les paramètres d'un modèle qui s'ajuste le mieux possible aux données et qui est capable de généraliser sur des nouvelles données non vues auparavant. Le cœur du problème revient à savoir comment estimer ces paramètres. L'approche probabiliste et l'apprentissage automatique offrent plusieurs choix pour le résoudre, chacun avec ses propres motivations théoriques et philosophiques : *Maximum-Likelihood*, modèles de Markov, réseaux de neurones... Enfin un mot sur l'intégration dans l'approche probabiliste. Il permet de diviser un modèle de langage en plusieurs couches : morphologie, syntaxe, sémantique et ainsi de suite. Accepter cette division implique que ces couches peuvent fonctionner séparément les unes des autres. C'est à dire, que la syntaxe et la morphologie sont totalement déconnectées, ce qui est faux, spécialement dans des langues sémitiques (Sima'an, 2003). La question de savoir comment résoudre cette intégration est un problème difficile pour la linguistique. Une possible solution se trouve dans l'utilisation de la définition des probabilités conditionnelles. C'est cela d'ailleurs qui permet, par exemple, d'utiliser le modèle du canal bruité pour la traduction statistique ou encore l'analyse sémantique, une fois que l'analyse syntaxique a été réalisée.

Après cette réflexion, on peut se poser les questions suivantes : *La linguistique peut-elle encore jouer un rôle dans le traitement de la langue naturelle ? Les méthodes numériques sont elles complètement adéquates et suffisantes pour les tâches du TAL ?* Je réserve mes réponses pour le chapitre de Bilan général et Perspectives.

On ne sait pas encore écrire des programmes qui comprennent le texte tel que le fait un être humain (Ibekwe-SanJuan, 2007). Cela devient trop complexe. Il est parfois difficile pour un individu d'expliquer comment il arrive à extraire des conclusions à partir d'une lecture entre les lignes. Il faut aborder le problème sous une autre optique, plus pragmatique peut-être. On ne peut pas, avec un logiciel, reproduire exactement la manière dont les personnes lisent et produisent des documents. Ce sont des chemins très différents, séparés par un abîme d'expériences, de vécus, des perceptions de la réalité... La démarche de la machine reste donc très inhumaine. Et c'est là où réside, entre autres, sa force. Une démarche numérique du TAL, quoi qu'on en dise, reste très compréhensible, car tout est réduit à des chiffres ou des probabilités qui sont assez parlantes. On peut à tout moment savoir pourquoi un système a privilégié tel choix par rapport à autre. À mon avis, on n'a probablement pas besoin d'écrire des programmes qui comprennent véritablement le texte. Il est connu, par exemple, que les résumeurs professionnels n'ont pas besoin de comprendre un document pour en rédiger un résumé pertinent : il leur suffit d'en avoir la technique. On a besoin uniquement d'écrire des programmes qui *raisonnablement* traitent les masses de documents à la place des personnes... et qu'ils le fassent rapidement. Tant que la démarche reste efficace, peu importe si elle est inhumaine, de près ou de loin pourvu que le résultat soit au rendez-vous en termes de quantité et de qualité.

Je commencerai par les perceptrons et les règles d'apprentissage basées sur la physique statistique. Puis la classification de courriels, d'opinions, et les méthodes probabilistes pour identifier auteur et thématique. Je présenterai un système de résumé automatique, une application au raffinement des requêtes, pour revenir enfin à la physique

statistique en introduisant le concept d'énergie textuelle comme une nouvelle mesure de similarité.

Chapitre 1

Classification d'objets par des perceptrons

Si l'homme est neuronal, le neurone lui,
est certainement inhumain

Au milieu des années 90 j'ai eu la chance de connaître Mme Mirta Gordon, à l'époque directrice de Recherche CNRS détachée au Commissariat à l'Énergie Atomique (CEA) de Grenoble. J'ai passé ainsi mon stage de DEA puis mon doctorat sous sa direction, dans le Laboratoire Magnétisme et Diffraction Neutronique (MDN), chez les physiciens théoriciens du CEA. J'ai discuté avec plusieurs d'entre eux, ce qui m'a fait connaître un monde inconnu pour moi, mais tout à fait fascinant. Des discussions scientifiques avec M. Peretto et M. Grempel, en plus de Mme Gordon, m'ont conduit peu à peu à constater que l'informatique, d'où je venais, et la physique statistique, avaient des liens très forts. J'étais doublement confronté au fait de comprendre les outils de la physique et de les adapter à l'informatique, plus particulièrement aux problèmes de la classification automatique supervisée. Selon moi, le mot *Neutronique* rimait bien avec *Neuronique*, donc neurones. Les réseaux de neurones ont donc façonné ma vision de la réalité. Je commencerai ce chapitre par la classification d'objets par des perceptrons, en m'appuyant sur un algorithme que j'ai bien étudié et modifié dans ma thèse : l'algorithme Minimerror. Cet algorithme, basé sur la physique statistique, permet de trouver les poids optimaux d'un perceptron binaire. Les perceptrons sont, cependant, très limités quant à leur capacité de séparation (donc de classification) d'objets. Une approche de réseaux de neurones multicouches s'avère donc nécessaire. Mais les problèmes de l'architecture et des unités cachées ne sont pas évidents. Dans ma thèse j'ai suivi une démarche incrémentale : ainsi, une fois établie qu'un réseau monocouche est capable de réaliser un grand nombre de fonctions des entrées (à condition d'avoir un nombre suffisant d'unités) (Hornik et al., 1989), on laisse évoluer l'architecture dynamiquement en fonction de la complexité du problème. Cette heuristique s'appelle Monoplan. Je présente dans ce chapitre la suite de ces algorithmes, une étude de la parité N-dimensionnelle, la proposition d'une version orientée à l'apprentissage non supervisé et son application dans un problème de détection de minéraux.

1.1 Apprentissages supervisé et non supervisé

Pour une tâche de classification, l'apprentissage est supervisé si les étiquettes des classes des patrons sont données *a priori*. Une fonction de coût calcule la différence entre les sorties souhaitées et réelles produites par un classifieur, puis, cette différence est minimisée en modifiant les paramètres par une règle d'apprentissage. En particulier, si le classifieur est un réseau de neurones, les paramètres sont l'ensemble des poids et son architecture. Un ensemble d'apprentissage supervisé est constitué par P couples, tel que $\mathcal{L} = \{(\vec{\xi}^{\mu}, \tau^{\mu}), \mu = 1, ..., P, \tau^{\mu}\}$; où $\vec{\xi}^{\mu}$ est le patron d'entrée μ, et τ^{μ} sa classe. En particulier $\tau^{\mu} \pm 1$ dans le cas des problèmes à deux classes. $\vec{\xi}^{\mu}$ est un vecteur de dimension N avec des valeurs numériques ou catégoriques. Si les étiquettes τ^{μ} ne sont pas présentes en $\mathcal{L}$, il s'agit d'apprentissage non supervisé : la classe des objets n'est pas connue et on cherche à établir des similarités entre les patrons.

Dans ce mémoire, lors des expériences, je ferai appel aux deux types d'apprentissage, séparément ou ensemble. Dans une autre approche intermédiaire, dite semi-supervisée, le classifieur nécessite de connaître un petit sous-ensemble des exemples d'apprentissage $\mathcal{L}$ avec leur classe, afin d'effectuer des regroupements sur le reste des données. Je ferai appel à cette stratégie dans les tâches de classification de courriels (c.f. section 2.2).

Le neurone formel (Hertz et al., 1991) peut se trouver dans l'état actif (sa sortie $\sigma = 1$) ou inactif (sa sortie $\sigma = 0$). Celui-ci additionne les signaux reçus, qu'on note $\vec{\xi} = (\xi_1, \cdots, \xi_N)$, qui sont pondérés par des poids $\mathbf{w} = (w_1 \cdots, w_N)$. Si le résultat de cette sommation est supérieur au seuil $\theta = -w_0$, le neurone est actif, inactif autrement. La sortie d'un neurone peut s'écrire comme une fonction du champ h donné par :

$$h = \sum_{i}^{N} w_i \xi_i + w_0 = \mathbf{w} \cdot \vec{\xi} + w_0 \tag{1.1}$$

L'état du neurone formel est $\sigma = \Theta(h)$ où Θ est la fonction signe[1] :

$$\Theta(x) \equiv \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{autrement} \end{cases} \tag{1.2}$$

Dans la suite j'appellerai neurones binaires les neurones à deux états et je noterai leur sortie σ. Les poids $\mathbf{w} = (w_1 \cdots, w_N)$ et le biais w_0, appelés également paramètres du neurone sont obtenus par apprentissage. Les neurones utilisés dans les réseaux connectés en couches calculent une fonction non linéaire de leur champ.

1.2 Minimerror : un algorithme optimal de classification

Minimerror est un algorithme d'apprentissage optimal (Gordon et Berchier, 1993; Raffin et Gordon, 1995; Gordon et Grempel, 1995) pour des perceptrons. Cet algorithme

[1] On aurait pu aussi utiliser la fonction de Heaviside.

fait une recherche en gradient des poids normalisés $\vec{w}$, $\vec{w} \cdot \vec{w} = N$, à travers de la minimisation d'une fonction de coût paramétrisée,

$$E = \frac{1}{2} \sum_{\mu=1}^{P} V \left(\frac{\tau^{\mu} \vec{w} \cdot \vec{\xi}^{\mu}}{2T\sqrt{N}} \right) \tag{1.3}$$

$$V(x) = 1 - \tanh(x) \tag{1.4}$$

Le paramètre T, appelé température (pour des raisons liées à l'interprétation de la fonction de coût), définit une largeur de la fenêtre efficace des deux côtés de l'hyperplan séparateur. La dérivée $\frac{dV(x)}{dx}$ est petite en dehors de cette fenêtre. Par conséquent, si le coût minimum (1.3) est recherché par une descente en gradient, seulement les patrons μ à une distance

$$|\gamma^{\mu}| \equiv \frac{|\vec{w} \cdot \vec{\xi}^{\mu}|}{\sqrt{N}} < 2T \tag{1.5}$$

contribueront de manière significative à l'apprentissage (Raffin et Gordon, 1995). L'algorithme Minimerror réalise cette minimisation commençant à une température élevée. Les poids sont initialisés avec la règle de Hebb, qui est le minimum de (1.3) dans la limite à températures élevées. Puis, T est diminuée lentement sur les itérations successives de la descente en gradient —*le recuit déterministe*— de sorte que seulement les patrons à l'intérieur de la fenêtre de largeur 2T sont effectivement pris en compte pour calculer le δ correctif :

$$\delta \vec{w} = -\epsilon \frac{\partial E}{\partial \vec{w}} \tag{1.6}$$

à chaque itération, où ϵ est la vitesse d'apprentissage. Ainsi, la recherche de l'hyperplan devient de plus en plus local à mesure que le nombre d'itérations augmente. Dans la pratique, on a constaté que la convergence est considérablement accelerée si les patrons déjà appris sont considérés à une plus basse température T_L que celle des patrons non appris, $T_L < T$. L'algorithme Minimerror a trois paramètres libres : la vitesse d'apprentissage ϵ de la descente en gradient, le rapport de température T_L/T, et la vitesse du recuit δT à laquelle la température diminue. À la convergence, une dernière minimisation avec $T_L = T$ est exécutée. Minimerror est un algorithme qui a des bonnes performances en problèmes de grande dimensionalité (Torres Moreno et Gordon, 1998a,b; Torres-Moreno et al., 2002), tel que je l'ai montré dans la solution du problème d'echos de sonar (Gorman et al., 1988). Plusieurs efforts (Berthold, 1996; Berthold et al., 1995; Bruske et Sommer, 1995; Chakraborty et Sawada, 1996; Karouia et al., 1995) n'ont pas réussi à découvrir qu'il est pourtant linéairement séparable ! (Torres Moreno et Gordon, 1998b; Hasenjäger et Ritter, 1999; Perantonis et Virvilis, 1999). Grâce à la puissance de Minimerror, nous avons été les premiers à le découvrir.

1.3 Monoplan : un réseau qui grandit en apprenant

Afin de résoudre des problèmes non linéairement séparables, il est nécessaire d'utiliser un réseau de neurones multicouches. Une approche constructive contrôle la croissance du réseau (nombre d'unités) selon la difficulté de l'ensemble d'apprentissage,

au contraire à la Rétropropagation de l'erreur (BP) et ses variantes (Peretto, 1992), qui supposent une architecture fixée à l'avance. Dans ma thèse j'ai introduit l'algorithme Monoplan (Torres Moreno, Juan-Manuel, 1998; Torres Moreno et Gordon, 1995) et ses variantes NetLS et NetSphères, où chaque unité cachée ajoutée corrige les erreurs d'apprentissage commises par l'unité précédente. Minimerror a été couplé avec ces heuristiques incrémentales (Torres Moreno, Juan-Manuel, 1998; Gérard et al., 2002). Plusieurs résultats dans (Torres Moreno, Juan-Manuel, 1998; Torres Moreno et Gordon, 1998a,b), ou plus récemment (Godin, Christelle, 2000) montrent que la méthode incrémentale est très puissante et donne des erreurs de généralisation comparables à d'autres méthodes. Un résumé de cet algorithme est :

- COUCHE CACHÉE. Un perceptron entraîné avec Minimerror apprend l'ensemble d'apprentissage $\mathcal{L}$. Si le nombre d'erreurs est nul, $\varepsilon_t = 0$, alors $\mathcal{L}$ est linéairement séparable et on stoppe l'algorithme : le réseau sera un perceptron simple. Si $\varepsilon_t > 0$, ce perceptron devient la première unité cachée, $h = 1$. Une deuxième unité $h+1$ est ajoutée, et les classes à apprendre sont modifiées : $\tau^{\mu}_{h+1} = \sigma^{\mu}_{h}\ \tau^{\mu}_{h}$ ($\tau_{h+1} = +1$ pour les patrons bien classés et $\tau_{h+1} = -1$ pour ceux qui ne le sont pas). On a montré que chaque perceptron est capable de corriger au moins une erreur d'apprentissage commise par le perceptron précedent. Ceci garantit la convergence de l'algorithme (Martinez et Estève, 1992; Gordon, 1996). Dès que l'apprentissage du perceptron h est finit, ses poids sont gèlés. La couche cachée est augmenté jusqu'à ce que la dernière unité apprenne correctement toutes les sorties.
- COUCHE DE SORTIE. L'unité de sortie ζ est connectée aux unités de la couche cachée. Cette unité apprend les sorties souhaitées τ^{μ}. Si les représentations internes sont LS, ζ les apprendra et l'algorithme sera stoppé. Autrement, on retourne à la phase d'agrégation des unités cachées, mais les sorties à apprendre par la nouvelle unité cachée $h+1$ seront : $\tau^{\mu}_{h+1} = \tau^{\mu}\ \zeta^{\mu}$

Ces deux phases convergent, comme montré par (Torres Moreno et Gordon, 1998a). Monoplan éngendre une machine de parité dans la couche cachée : les sorties réalisent la n-parité des représentations internes (Martinez et Estève, 1992; Biehl et Opper, 1991). Cependant, contrairement à l'algorithme *Offset* (Martinez et Estève, 1992) qui utilise une deuxième couche cachée pour calculer la parité (si le neurone de sortie détecte que les représentations internes ne sont pas linéairement séparables) Monoplan augmente la dimension de la couche cachée jusqu'à ce que les représentations internes soient linéairement séparables. Monoplan a fait ses preuves (Torres Moreno et Gordon, 1995, 1998a) en montrant des performances équivalentes à d'autres méthodes, mais en utilisant moins de paramètres. (Godin, Christelle, 2000) a developpé NetLS, une heuristique combinant des neurones linéaires et sphériques, telle que je l'avais évoqué dans ma thèse, ce qui confirme mes résultats.

1.4 La parité N-dimensionnelle et ses pièges

Le problème de la parité N-dimensionnelle a été étudié pendant longtemps par la communauté de réseaux de neurones. Minsky et Papert (Minsky et Papert, 1969) ont démontré, d'une manière élégante, qu'un perceptron est incapable de résoudre des pro-

blèmes tels que la parité à 2 entrées (ou en générale, le problème ou-exclusif XOR). La capacité d'un simple perceptron est limitée, puisqu'il peut résoudre uniquement les problèmes linéairement séparables (Cover, 1965; Gardner, 1987). La N-parité est difficile même dans de petites dimensions : $N \leq 5$, et sa difficulté augmente exponentiellement en fonction du nombre de patrons disponibles. Ce problème a été abordé par plusieurs méthodes, telles que la BP et ses variations. Ces méthodes ont des difficultés même dans de petites dimensions dues aux minimums locaux dans lesquels la minimisation de la fonction de coût peut être piegée. Cette anomalie est provoquée par le grand nombre d'états voisins (dans l'espace des entrées leurs distances de Hamming sont $d_H = 1$) avec des sorties opposées. Une solution avec des poids binaires a été présentée par (Kim et Park, 1995). Une approche alternative est l'utilisation de réseaux de neurones incrémentaux qui ajoutent des unités afin de corriger les erreurs d'apprentissage avec une heuristique appropriée, comme je l'ai montré dans ma thèse (Torres Moreno, Juan-Manuel, 1998).

L'étude concernant la solution de la N-parité je l'ai commencé au CEA/Grenoble avec Mirta Gordon, puis je l'ai continué avec Julio Aguilar au LANIA, Mexique. L'ensemble de résultats a été publié dans la revue *Neural Processing Letters* (Torres-Moreno et al., 2002).

1.4.1 Le nombre minimum de fautes

Le problème de la parité N-dimensionnelle est formulé comme un problème d'apprentissage supervisé avec un ensemble d'apprentissage $\mathcal{L}$ de P patrons à N entrées binaires et une sortie binaire. Le classifieur, en séparant les patrons de la classe $\tau = +1$ de ceux de $\tau = -1$, agit sur un problème d'apprentissage exhaustif car tous les $P = 2^N$ patrons doivent être appris. L'espace des entrées pour la parité N-dimensionnelle et les séparateurs $\vec{w}$ sont representés pour $N = 2, 3, 4$ dans la figure 1.1. La N-parité est devenue un défi classique pour les algorithmes de classification car elle est un problème fortement non linéairement séparable (non LS).

Selon une approche constructive, un réseau de neurones incrémental ajoute des unités cachées une à une, jusqu'à ce qu'il soit capable d'éliminer les erreurs d'apprentissage. Dans la parité N-dimensionnelle, le problème est difficile pour la première unité, car elle doit trouver le nombre le plus petit de fautes d'apprentissage dans un espace N-dimensionnel fortement imbriqué. Cependant, si le premier hyperplan est bien localisé, il réalise la solution du *minimum nombre d'erreurs* et en outre, grâce à la symétrie géométrique du problème, il sera plus facile à résoudre pour les unités suivantes.

Mais quel est le *nombre minimum d'erreurs* pour la parité N-dimensionnelle ? Pour trouver théoriquement ce nombre, d'abord, il faut considérer la figure 1.1 représentant la 2-parité. Le vecteur $\vec{w}$ sépare l'espace d'entrées, où on l'observe que les patrons $\mu = 2, 3$ et 4 sont bien classés, tandis que celui de classe négative $\mu = 1$ ne l'est pas. $\vec{w}_1$ fait une erreur de classification et il est impossible de faire mieux avec un perceptron simple. Pour la $3-$parité, dans la même figure, le vector $\vec{w}_1$ classe les patrons $\mu = \{1, 2, 3, 6, 7, 8\}$ correctement, tandis que les patrons $\mu = \{4, 5\}$ ne le sont pas. Donc deux

erreurs sont commises. On observe également le phénomène symétrique d'alternance des signes à partir de la position de l'hyperplan séparateur : les patrons avec $\tau^\mu = -1$ ($\mu = 5$), $\tau^\mu = +1$ ($\mu = 1,3,7$), $\tau^\mu = -1$ ($\mu = 2,6,8$) et $\tau^\mu = -1$ ($\mu = 4$). Un espace 4-dimensionnel est représenté dans $\mathbb{R}^2$, produisant un hypercube représenté sur la figure 1.1. L'hyperplan séparateur $\vec{w}$ classe mal cinq patrons. Une distribution de signes symétrique des patrons dans l'espace d'entrées est évidente, ce qui nous fait penser à un comportement combinatoire.

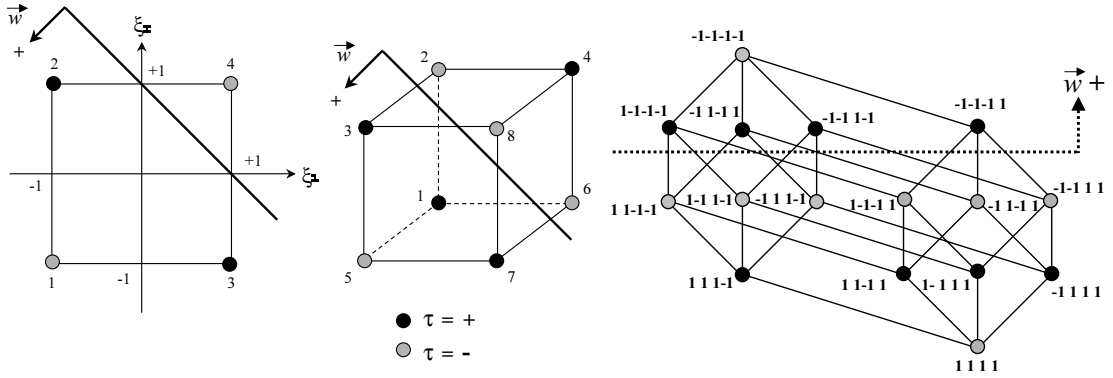


FIG. *1.1:* *Parité N-dimensionnelle avec $N = 2,3,4$.*

N		ν_0	ν_1	ν_2	ν_3	ν_4	ν_5	ν_6	ν_7	ν_8	ν_9	ν_{10}	ν_{11}	ν_f
2		1	2	1										1
3		1	3	3	1									2
4		1	4	6	4	1								5
5		1	5	10	10	5	1							10
6		1	6	15	20	15	6	1						22
7		1	7	21	35	35	21	7	1					44
8		1	8	28	56	70	56	28	8	1				93
9		1	9	36	84	126	126	84	36	9	1			186
10		1	10	45	120	210	252	210	120	45	10	1		386
11		1	11	55	165	330	462	462	330	165	55	11	1	772
	τ=	-	+	-	+	-	+	-	+	-	+	-	+	

TAB. *1.1:* *Distribution de sommets $\nu_k, k = 0,1,\cdots,N$, leur classe τ et le nombre minimum d'erreurs ν_f pour la parité $N = 1,2,\cdots,11$-dimensionnelle.*

Ces observations nous ont permis de construire la table 1.1, où la distribution de classes des sommets de l'hypercube ν_k est définie par les coefficients du binôme :

$$\nu_k = \binom{N}{k}; k = 0,1,\cdots,N \tag{1.7}$$

Cette table représente le triangle du Pascal. La distribution de classes alternée des sommets ν_k (la classe de patrons $\tau = -1$ ou $\tau = +1$) peut être séparée pour des hyperplans successifs. Si $N = 3$, nous avons un patron de la classe -1 (sommet ν_0), trois de la classe

$+1$ (sommet ν_1), trois de la classe -1 (sommet ν_2) et un patron de la classe $+1$ (sommet ν_3). L'hyperplan séparateur minimisant le nombre d'erreurs *devrait* être situé entre ν_1 et ν_2, ce qui génère 2 erreurs. Pour $N = 4$, on a un patron de la classe -1 (sommet ν_0), quatre de la classe $+1$ (sommet ν_1), six de la classe -1 (sommet ν_2), quatre de la classe $+1$ (sommet ν_3) et enfin un patron de la classe $+1$ (sommet ν_4). Pour réduire au minimum les erreurs de classification, l'hyperplan séparateur devrait maintenant se positionner entre ν_1 et ν_2 ou entre ν_2 et ν_3, où il produit 5 erreurs. La dernière colonne ν_f de la table 1.1 représente le nombre minimum d'erreurs pour la parité N-dimensionnelle realisé par un perceptron. ν_f n'est pas une simple sommation, car la parité des sommets doit être considérée. Une analyse géométrique a montré que :

$$\nu_f = \begin{cases} \nu_f(N=2p) & = \sum_{i=1}^{p} \binom{2p}{2p-i+1} \quad \text{si } N \text{ est pair} \\ \nu_f(N=2p+1) & = 2\nu_f(2p) \quad \text{si } N \text{ est impair} \end{cases} \qquad p = 1, 2, 3, \cdots \tag{1.8}$$

Nous introduisons ici le théorème suivant :

Théorème 1.4.1 *Soit $\mathcal{L} = \{(\vec{\xi}^{\mu}, \tau^{\mu}), \mu = 1, \cdots, P\}$ un ensemble d'apprentissage de P patrons binaires ξ_i^{μ} avec $P = 2^N; i = 1, 2, \cdots, N; \tau = \pm 1$ pour le problème de parité dans un espace d'entrées N-dimensionnelle. Le nombre minimum d'erreurs ν_f commis par un hyperplan séparateur optimal est donné par :*

$$\nu_f = 2^{N-1} - \binom{N-1}{m} \tag{1.9}$$

Démonstration : *Soit la distribution de classes des sommets de la parité N-dimensionnelle donné par :*

$$\nu_k = \binom{N}{k} \tag{1.10}$$

supposons que l'hyperplan séparateur soit placé entre ν_m et ν_{m+1}, orienté de telle manière que les patrons aux deux sommets ν_m et ν_{m+1} sont bien classés. Si m est pair, les patrons mal classés à gauche de l'hyperplan séparateur, selon le vecteur normal, sont dans les sommets $\nu_1, \nu_3, \cdots, \nu_{m-1}$. Si m est impair, alors les erreurs sont dans les sommets $\nu_0, \nu_2, \cdots, \nu_{m-1}$. On appele η_1 la première moitié du nombre d'erreurs, et on a :

$$\eta_1 = \begin{cases} \sum_{k=0}^{\frac{m}{2}-1} \binom{N}{2k+1} & \textit{si } m \textit{ est pair} \\ \sum_{k=0}^{\frac{m-1}{2}} \binom{N}{2k} & \textit{si } m \textit{ est impair} \end{cases} \tag{1.11}$$

$$= \sum_{k=0}^{m-1} \binom{N-1}{k} \tag{1.12}$$

De la même façon, on peut compter les erreurs η_2 du côté droit de l'hyperplan séparateur :

$$\eta_2 = \begin{cases} \sum_{k=2}^{\frac{N-m}{2}} \binom{N}{m+k} & \text{si } N-m \text{ est pair} \\ \sum_{k=2}^{\frac{N-m-1}{2}} \binom{N}{m+k} & \text{si } N-m \text{ est impair} \end{cases} \tag{1.13}$$

$$= \sum_{k=m+1}^{N} \binom{N-1}{k} \tag{1.14}$$

De ce fait, η_2 est la deuxième moitié du nombre d'erreurs. Étant donné que $v_f = \eta_1 + \eta_2$, on a :

$$\begin{aligned} v_f &= \sum_{k=0}^{m-1} \binom{N-1}{k} + \sum_{k=m+1}^{N} \binom{N-1}{k} \\ &= \sum_{k=0}^{N} \binom{N-1}{k} - \binom{N-1}{m} \end{aligned} \tag{1.15}$$

alors :

$$v_f = 2^{N-1} - \binom{N-1}{m} \tag{1.16}$$

Et donc, v_f sera plus petit quand $\binom{N-1}{m}$ est plus grand. En outre, si $N = 2p$, alors $m = p$, et si $N = 2p+1$ alors $m = p$ ou $m = p+1$. Q.E.D.

1.4.2 La solution de Minimerror

J'ai décidé de vérifier l'expression (1.9) expérimentalement. Pour ce faire, j'ai préparé des ensembles d'apprentissage exhaustifs de la parité N-dimensionnelle avec $2 \leq N \leq 15$ et $P = 2^N$. Dans tous le cas, la solution trouvé par Minimerror pour la première unité cachée correspond exactement au nombre d'erreurs prédits par (1.9). Pour le reste des unités, bien qu'il est connu que le nombre exact d'unités cachées nécessaires avec un réseau à une seule couche cachée sans rétroaction est $H = N$, la Retropropagation (et d'autres algorithmes non-constructifs (Peretto, 1992)) ne peut pas le trouver. Monoplan a trouvé la solution correcte, et elle a été vérifiée expérimentalement jusqu'à $N \leq 15$. Les résultats expérimentaux au delà de $N > 15$ sont très difficiles à obtenir, car le gradient cherche la minimisation d'un coût parmi un très grand nombre de mimimum locaux.

La dégénérescence des représentations internes

Il est possible que plusieurs patrons soient associés à la même représentation interne. En d'autres termes, quelques représentations internes sont dégénérées, puisqu'elles associent une représentation interne $\vec{\sigma}^{\mu}$ à plusieurs patrons μ. Par exemple,

i	Biais	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
1	-1.04	-1.10	0.52	1.00	1.03	-1.03	-1.07	-1.02	-1.00	-1.07	-1.00
2	1.44	-0.93	0.88	0.92	0.92	-0.96	-0.93	-0.97	-1.06	-0.93	-0.93
3	2.45	0.68	-0.69	-0.73	-0.71	0.68	0.72	0.74	0.73	0.71	0.68
4	2.47	-0.70	0.72	0.69	0.70	-0.70	-0.71	-0.69	-0.71	-0.68	-0.69
5	2.87	-0.54	0.54	0.51	0.53	-0.52	-0.52	-0.54	-0.50	-0.54	-0.52
6	2.84	0.49	-0.50	-0.58	-0.53	0.54	0.54	0.57	0.56	0.54	0.55
7	3.03	0.39	-0.37	-0.46	-0.45	0.41	0.42	0.43	0.47	0.42	0.45
8	3.06	-0.45	0.46	0.33	0.39	-0.42	-0.43	-0.40	-0.37	-0.43	-0.39
9	3.12	0.23	-0.17	-0.62	-0.40	0.26	0.19	0.31	0.49	0.25	0.39
10	3.12	-0.49	0.63	0.17	0.22	-0.28	-0.42	-0.33	-0.22	-0.38	-0.15

TAB. *1.2:* *Poids des couches cachées pour la parité* 10*-dimensionnelle*

dans le problème du XOR les quatre patrons sont associés seulement à trois états différents σ. Ceci est un phénomène souhaitable qu'on a appelé la contraction de l'espace d'entrées (Torres Moreno, Juan-Manuel, 1998). En effet, pour les P patrons appartenant à $\mathcal{L}$, seulement $P_\ell \leq P$ auront des représentations internes $\vec{\sigma}^\nu$; $\nu = 1, \cdots, P_\ell$.

Du point de vue du perceptron de sortie, il suffit d'apprendre les P_ℓ différentes représentations internes et de négliger celles répétées, c'est-à-dire, dégénérées. Expérimentalement nous avons trouvé qu'un grand nombre de représentations internes répétées peuvent compliquer (voire même empêcher) le positionnement correct de l'hyperplan séparateur au niveau du neurone de sortie. En effet, si une représentation interne est très dégénérée, elle contribue à l'apprentissage avec un coefficient multiplié par sa dégénération (nombre de répétitions). Par exemple, au cas extrême où il y a seulement deux représentations internes différentes : σ^1 et σ^2, avec un exemple simple associé à σ^1, et $P-1$ exemples associés à σ^2, σ^2 est très dégénéré. Si P est très grand, la contribution de σ^1 pour apprendre ne sera pas très significative. Dans ce cas Minimerror mettra l'hyperplan près du σ^2, et il aura besoin d'une grande quantité d'itérations pour le placer exactement entre σ_1 et σ_2. Puisque deux représentations internes identiques sont fidèles, il est impossible qu'elles donnent des sorties différentes, pour apprendre la sortie il suffit de garder uniquement les représentations internes différentes. Ces représentations constituent l'ensemble d'apprentissage non dégénéré $\mathcal{L}_\ell = \{(\vec{\sigma}^\mu, \tau^\nu)\ ;\ \ \nu = 1, \cdots, P_\ell\}$ plus petit que $\mathcal{L}$. Ce procédé a l'avantage supplémentaire d'un apprentissage robuste. Les tables 1.2 et 1.3 montrent la solution complète de la 10-parité.

Biais	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
1.00	1.00	-1.00	1.00	1.00	-1.00	-1.00	1.00	1.00	-1.00	-1.00

TAB. *1.3:* *Les poids du perceptron de sortie pour la parité* $10-$*dimensionnelle*

1.5 Classification hybride : *Fuzzy-k* et perceptrons sphériques

En 2002, lors d'un séjour comme chercheur invité au Laboratoire Lorrain de Recherche en Informatique et ses Applications[2] (LORIA) de l'INRIA (équipe Cortex), on m'a proposé d'étudier le problème de classification de minéraux. J'ai decidé d'utiliser les perceptrons sphériques pour trouver une solution. Cette section présente une stratégie d'apprentissage hybride pour des tâches de classification non supervisées. J'ai combiné l'apprentissage *Fuzzy k-means* et la version sphérique de Minimerror pour développer une stratégie incrémentale permettant des classifications non supervisées.

Cette recherche a été réalisée en collaboration entre le Bureau des Recherches Géologique et Minière (BRGM)[3], l'équipe Cortex du LORIA-INRIA (France), l'École Polytechnique de Montréal et le Conseil de Recherche en Sciences Naturelles et Génie (CRSNG) (Grant Nb. 239862-01), Canada. Nous avons publié les résultats de cette étude dans le congrès ICANN/ICONIP 2003 (Torres-Moreno et al., 2003).

1.5.1 Expérience de prospection de minéraux

La division de ressources minérales du BRGM développe un système d'information géographique (GIS) à l'échelle du continent, concernant la recherche métallogénique. Ceci constitue un outil pour la prise de décisions. La compréhension de la formation de métaux tels que l'or, le cuivre ou l'argent n'est pas suffisante et il a beaucoup de modèles décrivant un site minier en incluant la taille du dépôt pour des métaux divers. Dans cette étude, nous nous concentrerons sur un GIS qui couvre la zonde des Andes et la classifie en deux classes : *site* minier et *barren*. Un *site* est une concentration minérale économiquement exploitable (Michel et al., 1964). Le facteur de concentration correspond au taux d'enrichissement d'un élément chimique, c'est-à-dire à la relation entre son contenu moyen dans l'exploitation et son abondance dans l'écorce terrestre. Les géologues opposent le concept de site à celui de *barren*. En fait, pour l'interprétation des résultats de généralisation, il est nécessaire de comptabiliser le nombre de sites bien classés dans chaque catégorie afin de répondre à la question : il s'agit d'un *site* ou d'un *barren* ? Dans notre étude, un *site* (représenté par un patron) sera consideré comme tel s'il contient au moins un métal et un *barren* par un endroit sans aucun métal. Ces classes seront utilisées dorénavant. Le GIS Andes contient 641 patrons, 398 du type *site* et 343 du type *barren*, codés sur 25 attributs, 8 qualitatifs et 17 quantitatifs. Ils correspondent à la position d'un site, le type et l'âge de la roche le contenant, la proximité du site à une faille caracterisée par son orientation dans une carte géographique, la densité et le profondeur focale des tremblements de terre juste au-dessous du site, la proximité des volcans actifs, la géométrie de la zone de subduction, etc. J'ai effectué une étude statistique pour déterminer l'importance de chaque variable. Pour chaque attribut, la moyenne des patrons *site* et *barren* a été calculés afin d'en déterminer quels pourraient être les plus discriminants pour les séparer (figure 1.2). Il y a quelques attri-

[2] http://cortex.loria.fr
[3] http://www.brgm.fr

buts (15, 16, 17 ou 22, parmi d'autres) qui ne semblent pas pertinents. D'autre part, les attributs 3, 5, 6 et 25 sont plutôt discriminants. Il est intéressant de savoir comment le choix des attributs influence l'apprentissage et particulièrement la tâche de généralisation. Pour cette raison, nous avons créé 11 bases de données avec différentes combinaisons des attributs. Le tableau 1.4 montre le nombre d'attributs qualitatifs et quantitatifs et la dimension des bases utilisée. La plage des valeurs étant extrêmement large, un

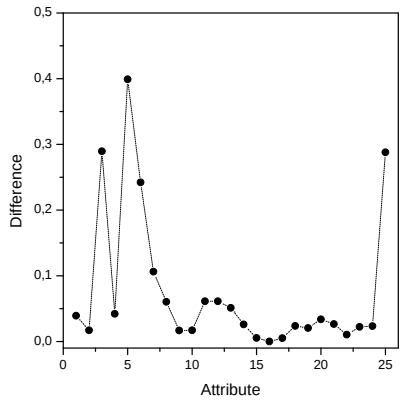


FIG. *1.2:* *Différences aux carrée de la moyenne des patrons moyens.*

Base de données	Attributs utilisés	Qual.	Quant.	*N*
I	1 à 25	8	17	25
II	1 à 8	8	0	8
III	9 à 25	0	17	17
IV	11,12,13,14	0	4	4
V	11,12,13,25	0	4	4
VI	3,5,6,7	4	0	4
VII	**11,12,13,14,25**	**0**	**5**	**5**
VIII	11,12,13,20,25	0	5	5
IX	3,5,6,7,11,12,13,25	4	4	8
X	11,12,13,14,18,19,20,21,23,24	0	10	10
XI	11,12,13,14,18,19,20,21,23,24,25	0	11	11

TAB. *1.4:* *Bases de données d'apprentissage du GIS Andes.*

pré-traitement des attributs quantitatifs s'impose afin de les homogénéiser. Ainsi, pour chaque variable continue, la standardisation calcule la moyenne et l'écart-type. Puis, la variable est centrée et toutes leurs valeurs ont été divisées par l'écart type. Les attributs qualitatifs ne sont pas modifiés. Le corpus ainsi modifié a été divisé en ensembles aléatoires allant de 10% (64 patrons) à 95 % (577 patrons) de la taille de la base originale (641 patrons). Le complément a été choisi comme ensemble de test.

1.5.2 L'algorithme non supervisé *Fuzzy k-means*

Fuzzy k-means (Bezdek, 1981; deGruijter et McBratney, 1988) permet d'obtenir un regroupement des éléments par une approche floue avec un certain degré d'appartenance, où chaque élément peut appartenir à une ou plusieurs classes, à différence de *k-means*, où chaque exemple appartient à une seule classe (partition dure). *Fuzzy k-means* minimise la somme des erreurs quadratiques avec les conditions suivantes :

$$\sum_{k=1}^{c} m_{\mu k} = 1; \quad \sum_{k=1}^{c} m_{\mu k} > 0; \quad m_{\mu k} \in [0,1]; \quad k = 1, \cdots, c; \quad \mu = 1, \cdots, P \tag{1.17}$$

La fonction objectif est définie par :

$$J = \sum_{\mu=1}^{P} \sum_{k=1}^{c} m_{\mu k}^{f} d^{\lambda}(\xi^{\mu}, \beta^{k}) \tag{1.18}$$

où P est le nombre de données dont on dispose, c est le nombre de classes désiré, β^k est le vecteur qui représente le centroïde (barycentre) de la classe k, ξ^μ est le vecteur qui représente chaque exemple μ et $d^{\lambda}(\xi^{\mu}, \beta^{k})$ est la distance entre l'exemple ξ^μ et β^k en accord avec une définition de distance qu'on écrira d_{uk}^{λ} afin d'alléger la notation. f est un paramètre, avec une valeur comprise dans l'intervalle [2,∞] qui détermine le degré de flou (*fuzzyfication*) de la solution obtenue *in fine*, contrôlant le degré de recouvrement entre les classes. Avec $f = 1$, la solution deviendrait une partition dure (du style *k-means*). Si $f \to \infty$ la solution approche le maximum de flou et toutes les classes risquent de se confondre en une seule. La minimisation de la fonction objectif J fournit la solution pour la fonction d'appartenance $m_{\mu k}$:

$$m_{\mu k} = \frac{d_{\mu k}^{\lambda/(f-1)}}{\sum_{j=1}^{c} d_{\mu j}^{\lambda/(f-1)}}; \; \beta^{k} = \frac{\sum_{\mu=1}^{P} m_{\mu k}^{f} \xi^{\mu}}{\sum_{\mu=1}^{P} m_{\mu k}^{f}}; \; \mu = 1, \cdots, P; \; k = 1, \cdots, c \tag{1.19}$$

1.5.3 Perceptrons hyperspèriques

Une variation de Minimerror, appelée Minimerror-S permet d'obtenir des séparations sphériques dans l'espace d'entrées. La séparation sphérique utilise la même fonction de coût (1.3) que la version linéaire, mais une stabilité sphérique définie par

$$\gamma_s = \left\| \vec{w} - \vec{\xi} \right\| - \rho^2 \tag{1.20}$$

où ρ est un rayon hypersphérique centré sur $\vec{w}$. La classe du patron est $\tau = -1$ à l'intérieur de la sphère et $\tau = 1$ à l'extérieur[4]. La complexité de ce perceptron reste la même

[4]La notion de distance d'un patron à l'interieur de la sphère au centre de la sphère ou à la frontière séparatrice n'est pas symétrique. Une transformation logarithmique peut être donc nécessaire.

que celui linéaire, le rayon prends la place du biais. Quelques tests ont suggéré que les perceptrons hypersphériques pouvaient être appliqués de façon efficace en problèmes de classification (Torres Moreno, Juan-Manuel, 1998; Torres Moreno et Gordon, 1998a). D'autres études (Godin, Christelle, 2000) ont approfondie les perceptrons sphériques avec Minimerror. Ainsi, des stratégies incrémentales combinant perceptrons linéaires et sphériques, NetLS (tel que je l'avais évoque dans ma thèse) ont confirmé les performances des réseaux incrémentaux.

1.5.4 Une stratégie hybride

J'ai choisi une stratégie combinée : une première couche cachée non supervisée calcule les centroïdes avec l'algorithme *Fuzzy k-means*. Comme entrée on aura P patrons à N entrées de l'ensemble d'apprentissage $\mathcal{L}$. Alors, Minimerror-S de façon supervisé trouve les séparations sphériques les mieux adaptées afin de maximiser la stabilité des patrons. L'entrée pour Minimierror est la même $\mathcal{L}$ ensemble, avec les étiquettes τ^{μ} calculées par *Fuzzy k-means*. Bien que le nombre de classes est choisi à l'avance, la couche non supervisée permet de se passer d'un étiquetage préalable des données, qui peut s'avérer assez coûteux dans le cas des sites miniers.

J'ai mesuré la performance en classification (pourcentage des sites bien classés) des ensembles de tests. La base **VII** (avec trés peu d'attributs quantitatifs) a eu les meilleures performances en apprentissage et en généralisation par rapport aux autres bases. En utilisant tous les attributs, les performances s'écroulent. La figure 1.3 à droite, montre quelques résultats de ce comportement. Basé sur cette information, j'ai gardé la base de données **VII** pour réaliser 100 essais aléatoires. La capacité de discrimination entre *site* et *barren*, selon le pourcentage des patrons appris est montrée sur la figure 1.3. La détection de la classe *site* est bien supérieur à celle de la classe *barren*. Une analyse fine des résultats a constaté que la détection d'or, d'argent et de cuivre restent très précises, mais, celle de molybdène est plutôt pauvre. Ceci peut être expliqué en accord avec la présence faible de ce métal. Dans les mêmes conditions, un perceptron multicouche avec 10 neurones sur une seule couche cachée obtient 77 % de bon classement.

1.5.5 Minimerror non supervisé ?

Après ma thèse, je me suis demandé s'il serait possible de créer une approche non supervisée de Minimerror. Les perceptrons sphériques, avec la stabilité γ_s définie par l'équation (1.20), sont de bons candidats pour créer un algorithme non supervisé. Une stratégie de croissance non supervisée a été suggérée pendant mon séjour au LORIA.

L'algorithme Minimerror-S non supervisé commence par obtenir les distances entre les patrons. Une distance euclidienne peut être employée pour les calculer. Une fois les distances établies, il commence par trouver la paire μ et ν de patrons avec la plus petite distance ρ. Ceci crée le premier noyau incrémental. On fixe le centre de l'hypersphère

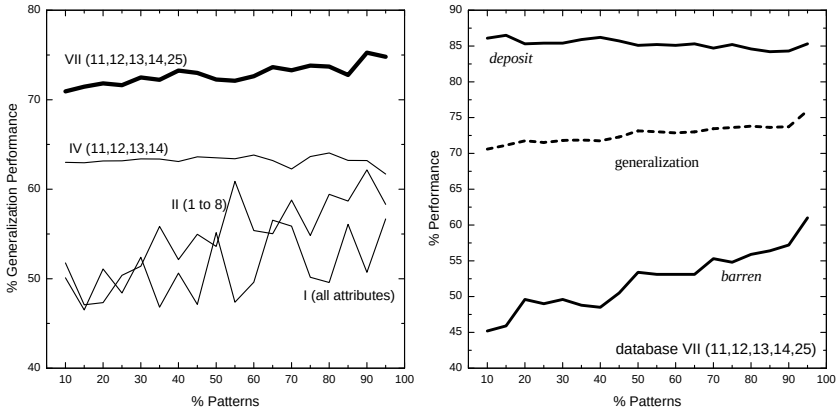


FIG. 1.3: *À gauche : performances de généralisation selon la taille de l'ensemble d'apprentissage obtenue par le modèle hybride avec les différentes bases de données. À droite : performances de généralisation* site/barren *selon la taille de l'ensemble d'apprentissage (100 tests) obtenues par le modèle hybride avec la base de données VII (reproduit de (Torres-Moreno et al., 2003), pag. 40)*

.

$\vec{w}_0$ au milieu des patrons μ et ν :

$$\vec{w}_0 = \frac{(\vec{\xi}^\mu + \vec{\xi}^\nu)}{2} \tag{1.21}$$

Le rayon initial est fixé

$$\rho_0 = \frac{3\rho}{2} \tag{1.22}$$

pour faire rentrer un certain nombre de patrons dans le noyau incrémental. Puis, les patrons sont marqués $\tau = -1$ s'ils sont à l'intérieur ou au bord de la sphère initiale, et $\tau = 1$ ailleurs. Minimerror-S trouve l'hypersphère $\{\rho*, \vec{w}*\}$ qui sépare le mieux les patrons. Les représentations internes sont $\sigma = -1$ si

$$-\frac{1}{\cosh^2(\gamma^\mu)} < \frac{1}{2}$$

autrement $\sigma = 1$. Ceci permet de vérifier s'il y a des patrons avec $\tau = 1$ à l'extérieur mais suffisamment proches de la sphère $(\rho_1^*, \vec{w}_1^*)$. Dans ce cas, $\tau = -1$ pour ces patrons et il les reapprend, répétant la procédure pour tous les patrons de $\mathcal{L}$. En même temps, il passe à un autre noyau de croissance qui formera une deuxième classe $\vec{w}_2$, calculant avec Minimerror-S $(\rho_2^*, \vec{w}_2^*)$, et répétant la procédure jusqu'à ce qu'il n'y ait plus de patrons à classer. Il obtient enfin k classes. Une procédure de réduction peut éviter d'avoir trop de classes en éliminant les perceptrons avec peu d'éléments (seuil fixé à l'avance). Il est possible de définir des conditions à la frontière, qui sont des restrictions qui empêchent de localiser le centre de l'hypersphère à l'extérieur de l'espace d'entrées. Pour certains problèmes cette stratégie peut être intéressante. Ces restrictions sont cependant facultatives : s'il y a trop d'erreurs d'apprentissage, l'algorithme décide de les négliger, le centre et le rayon des sphères séparatrices peuvent diverger.

J'ai testé la version de Minimerror non supervisé sur le GIS Andes. Malgré sa simplicité séduisante, le nombre de classes obtenues est parfois trop élévée. Les performances sont donc inférieures à celles obtenues avec l'approche hybride combinant *k-means*. Des reflexions plus approfondies doivent être faites.

1.6 Au délà des perceptrons

Les machines a support vectoriel (SVM) proposées par Vapnik (Vapnik, 1982, 1995) permettent de construire un classifieur à valeurs réelles qui découpe le problème de classification en deux sous-problèmes : transformation non-linéaire des entrées et choix d'une séparation linéaire *optimale*. Les données sont d'abord projetées dans un espace de grande dimension H muni d'un produit scalaire $\langle \cdots \rangle$ où elles sont linéairement séparables selon une transformation basée sur un noyau linéaire, polynomial ou gaussien. Puis dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui déterminent un hyperplan séparant correctement toutes les données et maximisant la *marge*, la distance du point le plus proche à l'hyperplan.

Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel (c'est-à-dire, trouver une hypothèse h pour laquelle la probabilité que h soit fausse sur un exemple non-vu et extrait aléatoirement du corpus de test soit minimale).

En pratique, cette transformation est implicite dans le noyau $K(x, y) = \langle \phi(x), \phi(y) \rangle$. La frontière de décision est de la forme :

$$f(x) = \langle w, \phi(x) \rangle_H + b = \sum_{\mu=1}^{P} \alpha_i K(x_i, x) + b \tag{1.23}$$

Dans le cas où les données sont linéairement séparables, la frontière $f(x)$ optimale (qui maximise la marge entre les classes) est obtenue en résolvant le problème de programmation quadratique sous contraintes :

$$min_w \frac{1}{2} ||w||^2 \tag{1.24}$$

s.a

$$y_i(< w, \phi(x_i) > + b) \geq 1 \forall i \in 1, \cdots, P \tag{1.25}$$

La complexité d'un classifieur SVM va donc dépendre non pas de la dimension de l'espace des données, mais du nombre de vecteurs de support nécessaires pour réaliser la séparation (Vapnik, 1995).

1.7 Conclusion

La classification d'objets est une tâche qui peut être réalisée par des perceptrons. J'ai décidé d'étudier en profondeur les capacités de la règle d'apprentissage Minimerror et

de l'algorithme Monoplan, qui grandit en apprenant. Pour cela j'ai étudié théoriquement le problème de la parité N-dimensionnelle et une application de classification des sites miniers.

La N-parité est une tâche de classification très difficile pour les réseaux de neurones. Nous avons trouvé une expression théorique pour obtenir le nombre minimum d'erreurs v_f comme fonction de N qui devrait correspondre à une séparation optimale. On a vérifié expérimentalement cette quantité pour $N = 1, \cdots, 15$ à l'aide d'un perceptron entraîné par Minimerror. On a aussi résolu le problème complet de la parité N-dimensionnelle selon une approche constructive avec un réseau de neurones *feed-forward* minimal à une seule couche cachée de $h = N$ unités. La combinaison hybride de Minimerror et de *Fuzzy K-means* semble être prometteuse du point de vue d'un apprentissage hybride. Cette stratégie a été appliquée à une base de données réelle, qui nous a permis de prévoir d'une manière plutôt satisfaisante si un site pouvait être identifié comme un dépôt de minerai ou pas. 75% des patrons bien classés par cet algorithme est comparable aux valeurs obtenues avec d'autres méthodes supervisées classiques. Ceci montre également la capacité discriminatoire des attributs descriptifs que nous avons choisi en tant que plus appropriés pour ce problème. Je pense qu'on devrait obtenir une amélioration significative des performances en augmentant le nombre d'exemples. Des études additionnelles doivent être effectuées pour déterminer plus exactement d'autres attributs pertinents, afin d'attaquer des problèmes multiclasses. J'ai suggéré également une variation de Minimerror pour la classification non supervisée, en utilisant des séparateurs hypersphériques. Cependant il faudra approfondir la stratégie pour la rendre plus performante.

D'autres méthodes plus avancées que les réseaux de neurones, telles que les SVM ont été aussi présentées. J'utiliserai l'ensemble de ces classifieurs pour des tâches de Traitement Automatique de la Langue Naturelle : catégorisation de texte, routage de courriels, résumé automatique... combinées avec d'autres techniques probabilistes comme les chaînes de Markov. Cela fera l'objet des deux chapitres suivants.

Chapitre 2

Classification de textes et le modèle vectoriel

C'est que l'écriture, Phèdre, a, tout comme la peinture, un grave inconvénient.
Les œuvres picturales paraissent des êtres vivants ;
mais, si tu les interroges, elles gardent un majesteux silence.
Il en est de même des discours écrits. Tu croirais que ce qu'ils disent, ils y pensent ;
mais, si tu veux leur demander de t'expliquer ce qu'ils disent,
ils te répondent toujours la même chose. Platon, Phèdre. [1]

En 1999, M. Jean-Guy Meunier, directeur du LANCI, m'a proposé un séjour post-doctoral en classification automatique de textes. Le texte, bien qu'intéressant pour moi pour plusieurs raisons (même dans le sens purement littéraire), n'a jamais été l'objet de mes recherches précédentes. J'ai donc accepté sans hésitation. Il m'a semblé naturel de vouloir explorer l'information textuelle sous l'optique des algorithmes connexionnistes et de classification que j'ai déjà maîtrisé. J'ignorais complètement la problématique des textes, de leur représentation et des outils existants pour le TAL. L'étendue du Web et de la masse de textes disponible m'était, sinon étrange, lointaine. Vite j'ai compris que le problème proposé par M. Meunier n'était pas évident, qu'on pouvait être noyé par la quantité de documents existants, et que leur traitement demandait des algorithmes performants et adaptés à l'univers textuelle et ses particularités. À mon avis il a un paradoxe avec le textuel : les mots sont figés, les phrases statiques, la dimension « temps » peut être réversible, on peut lire et relire le texte... et pourtant il y a plusieurs choses qui échappent à la compréhension automatique du texte. Il y a l'ambiguïté à plusieurs niveaux, du lexique à la sémantique, et au-delà, bien entendu, des barrières de la langue. Il y a aussi le fait que la quantité de textes, pouvant être générés avec un alphabet fini, soit infinie. Car le texte essai d'exprimer, en même temps, la réalité et l'imaginaire. Il est donc nécessaire de modéliser, de synthétiser ou de re-écrire cette expressivité, à la fois figée et changeante.

[1] *Los textos Fedro, como la pintura tienen un gran inconveniente. Las obras de pintura parecen seres vivos, pero si les preguntas algo, mantienen un silencio solemne. Lo mismo ocurre con las palabras escritas ; puedes suponer que entienden lo que dicen, pero si les preguntas lo que quieren decir, simplemente contestan con la misma respuesta una y otra vez.*

2.1 La vectorisation de textes

La représentation vectorielle de textes (Salton et McGill, 1983; Salton, 1989), même si elle est très différente d'une analyse structurale linguistique, s'avère un modèle performant et rapide (Manning et Schütze, 1999). Les méthodes vectoriels ont, d'ailleurs, la propriété d'être assez indépendantes de la langue. Dans ce modèle de représentation, les textes sont traités comme des sacs de termes. Les termes peuvent être assimilés à des mots ou à des n-grammes. Cependant, les données résultantes seront très volumineuses, clairsemées et très bruitées. Ceci signifie que l'information pertinente ne constitue qu'une faible partie de l'ensemble total des données disponibles (Lebart, 2004). La vectorisation d'un document crée une matrice terme-segment où chaque case représente la fréquence d'apparition d'un terme dans une phrase (ou segment). Cette matrice peut être très volumineuse. C'est pourquoi, des mécanismes de filtrage et de lemmatisation s'avèrent indispensables afin de réduire la complexité du lexique.

Si l'on utilise des mots, les termes sont construits de la manière suivante : le contenu textuel du document est segmenté en mots, les mots creux ou vides de sens (conjonctions, articles, prépositions, etc.) sont éliminés, puis les mots pleins sont ramenés à leur racine. D'après (Jalam et Chauchat, 2002), si l'on utilise des n-grammes, on peut se passer de certains processus de normalisation.

La lemmatisation (même si plus coûteuse en temps) et non un simple *stemming*[2] s'avère plus adaptée pour le français (Namer, 2000), qui est une langue latine à fort taux de flexion. Elle consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier. Bien que brutale (Ibekwe-SanJuan, 2007) mais efficace, dans mes modèles j'ai decidé de pousser encore plus loin ce traitement, en proposant des familles de mots plutôt qu'une simple lemmatisation. Par famille de mots j'entends des mots morphologiquement *proches*. Ainsi les formes *chante, chantaient, chanté, chanteront*... et éventuellement *chanteur* et *chanteuses* seront ramenés au même terme CHANTER, avant de leur associer un nombre d'occurrences. Ce processus (semi-automatique) pourrait être automatisé avec des algorithmes appropriés (Bernhard, 2007) . On y travaille. Rien n'empêche cependant de combiner les familles de mots et le *stemming*, dans le souci de reduire le lexique. Ces processus permettent d'amoindrir la malédiction dimensionnelle[3] qui pose de très sérieux problèmes de représentation dans le cas des grandes dimensions. D'autres mécanismes de réduction du lexique peuvent aussi être déclenchés, comme la detéction des mots composés. Des expressions courantes (*par exemple, c'est-à-dire, chacun de*...), la ponctuation et les symboles tels que $, #, *, etc. peuvent aussi être supprimés.

Dans le modèle vectoriel, l'ensemble de données dont on dispose, appelé l'ensemble d'apprentissage, consiste en P documents. Nous dénoterons cet ensemble $\mathcal{S}$, de façon analogue à l'ensemble d'apprentissage definie dans la Section 1.1. Si l'ensemble n'est pas étiqueté, on disposera uniquement des P documents et d'aucune information concernant leur classe. Celle-ci peut être ignoré et les documents seront regroupés

[2] Qui est en général bien adapté à l'anglais, par exemple.
[3] *The curse of dimensionality !*

uniquement en fonction de leurs caractéristiques. Dans une approche supervisée, on aura des classes τ^{μ} associées aux documents. Le corpus est alors vectorisé en P vecteurs (documents) $\overrightarrow{\sigma}$ de N dimensions (taille du lexique). Dans la matrice fréquentielle $\mathcal{S}$, chaque composante σ_i^{μ} ; $i = 1, 2, \cdots, N$ contient la fréquence du terme i dans un document μ :

$$\mathcal{S} = \begin{pmatrix} s_1^1 & s_2^1 & \dots & s_i^1 & \dots & s_1^N \\ s_1^2 & s_2^2 & \dots & s_i^2 & \dots & s_2^N \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ s_\mu^1 & s_\mu^2 & \dots & s_\mu^i & \dots & s_\mu^N \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ s_P^1 & s_P^2 & \dots & s_P^i & \dots & s_P^N \end{pmatrix}, \quad s_i^{\mu} = \{0, 1, 2, \dots\} \tag{2.1}$$

La visualisation de la matrice $\mathcal{S}$ met en évidence des caractéristiques intéressantes des textes. Par exemple, un graphique mettant en abscisse le numéro du segment μ et en ordonnée la première apparition du terme i, génère une image du texte semblable à celle représentée sur la figure 2.1, à gauche. La densité elevée en bas du graphique indique la reutilisation du terme i parmi tous les segments. L'allure de la courbe indique la vitesse d'introduction des nouveaux termes : plus le vocabulaire d'un auteur est riche, plus la pente est élevée. Dans la même figure 2.1 à droite, j'ai affiché les courbes correspondantes aux trois corpus indiqués : « Le discours de la méthode », textes scientifiques de l'Inra et le Coran (version française). L'axe vertical a été normalisé afin de pouvoir les comparer en affichanut uniquement les 650 premiers segments. Il est évident que Descartes utilise un vocabulaire très riche : en peu de phrases il introduit un grand nombre de mots. Les textes de l'Inra (mélange hétérogène d'articles scientifiques) montrent une courbe moins prononcée. Enfin le Coran affiche la vitesse d'introduction de termes la moins élevée : trop parcimonieux et répétitif. J'avance ici l'hypothèse que la dérivée $\frac{\partial \text{Terme}_i}{\partial \text{Segment}^{\mu}}$ pourrait indiquer la vitesse d'introduction du lexique par un auteur. Il faut cependant une étude approfondie afin de la confirmer.

Classphères : un réseau de jouet

Les réseaux ART reposent sur l'auto-organisation des connaissances en structures qui tendent à résoudre le délicat dilemme stabilité-plasticité. La plasticité spécifiant la capacité du système à appréhender des informations nouvelles, et la stabilité, sa capacité à les organiser en structures stables. Cette théorie a donné naissance à plusieurs modèles : ART1 et ART2 (Carpenter et Grossberg, 1991; Carpenter et al., 1991b), fuzzy ART (Carpenter et al., 1991c), ARTmap (Carpenter et al., 1991a) et fuzzy ARTmap (Carpenter et al., 1992). Ces modèles, comme les cartes auto-organisatrices de Kohonen (Kohonen, 1982) et la méthode GAR (Alpaydin, 1990) appartiennent aux réseaux de neurones à apprentissage non supervisé, dont les poids des interconnexions codent les prototypes des classes. ART1 utilise des données binaires, ce qui le rend spécialement utile pour la classification textuelle. Le nombre final de classes dépend d'un

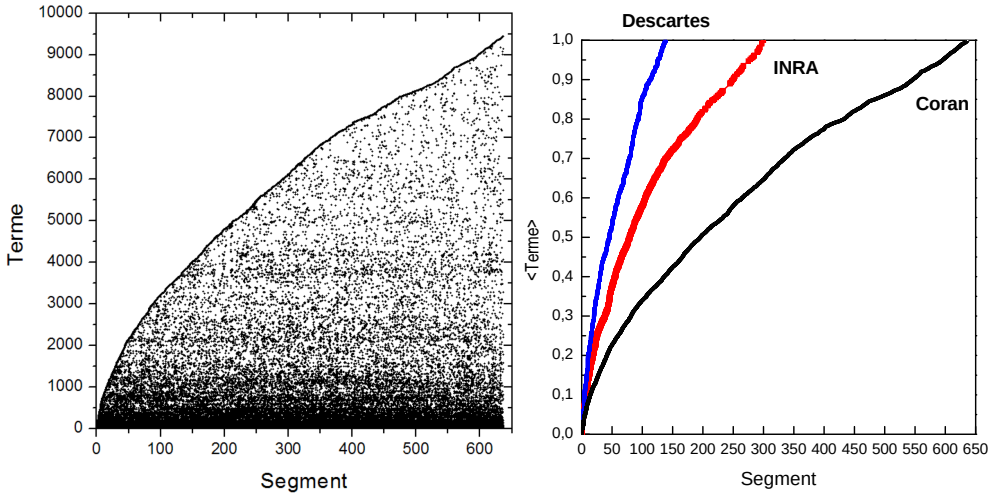


FIG. 2.1: *A gauche : Coran avec 637 segments et 9 438 mots, dont 2 524 termes. A droite : courbes normalisée de termes en fonction du segments, appartenant à différents textes.*

paramètre de vigilance $0 \leq \rho \leq 1$ fixé à l'avance. Plus ρ est proche de 1, plus les classes seront sélectives (moins d'éléments) et leur nombre important. Alors que pour des faibles valeurs de ρ, le nombre de classes sera faible, chaque classe comportant un grand nombre d'éléments. Bien que puissant pour des applications textuelles montrant des performances supérieures (Meunier et al., 1999) à d'autres méthodes, telles que *k-means*, ART1 est incapable de faire appartenir un même segment à plusieurs classes et dépend du choix de ρ. Au LANCI on utilisait le modèle vectoriel et ART1 pour classer les textes (Seffah et Meunier, 1995; Meunier et al., 1999).

En 2000, J. G. Meunier, Patricia Velázquez et moi même, avons développé au LANCI, l'algorithme incrémental Classphères (une sorte de perceptrons hypersphériques bien plus simples que ceux proposés dans la section 1.5.5), basé sur les distances entre hypersphères, pour essayer de palier les handicaps évoqués. Classphères est un algorithme à apprentissage non supervisé, qui construit des regroupements de segments qui se ressemblent en fonction d'une mesure de distance adéquate. Il trouve donc, les prototypes qui codent mieux les classes. Nous avons opté pour la version avec des distances de Hamming, car elle est mieux adaptée dans un espace binaire. On cherche tous les voisins les plus proches de chaque segment dans l'hypercube de dimension N, en les regroupant par rapport à une distance minimale. L'algorithme commence par créer la matrice triangulaire de Hamming : $H(\mu,\nu); 1 \leq \mu \leq P; \mu+1 \leq \nu < P$, qui correspond à la distance entre les segments μ et ν. Il suffit de calculer seulement la partie triangulaire supérieure car la matrice est symétrique. Puis, il construit le vecteur $V(\nu) : 2 \leq \nu \leq P$ qui est la distance minimale entre un segment ν fixé, et les segments μ, où $1 \leq \mu < \nu - 1$. Une classe est ainsi formée en cherchant dans $H(\mu,\nu)$ pour chaque ligne μ, les segments ν voisins qui ont la même distance minimale $V(\nu)$, où $\mu+1 \leq \nu < P$. Cet algorithme construit alors des hypersphères centrées sur un segment μ (patron), à rayon variable

$V(\nu)$, à l'intérieur desquelles on retrouve des segments voisins. Nous avons effectué plusieurs tests sur des textes de petite taille[4] (< 3000 mots) et de taille moyenne[5] (< 43 000 mots). Nous avons constaté que ART1 obtient un nombre plus important de classes avec des écarts type plus grands que Classphères (l'écart type augmente en fonction de la taille du texte pour les deux classifieurs). Bien que séduisant, nous avons calculé la complexité de Classphères comme $O(P^2)$, étant P le nombre de segments, ce qui limite son utilisation dans des corpus réels. Les résultats de ce réseau de jouets, ont été publiés dans les mémoires du congrès JADT 2000 (Torres-Moreno et al., 2000).

Les méthodes de fouille de textes, utilisant le modèle vectoriel, apportent des solutions partielles aux tâches de filtrage, de routage, de recherche d'information, de classification thématique et de structuration non supervisée. Ces méthodes présentent, de surcroît, l'intérêt de fournir des réponses adaptées à des situations où les corpus sont en constante évolution ou bien de se passer des barrières de la langue. Je présenterai dans les sections suivantes deux applications de classification textuelle basés sur le modèle vectoriel.

2.2 Routage automatique de courriels

Les nouvelles formes de communication écrite posent des défis considérables aux systèmes du TAL. On observe des phénomènes linguistiques bien particuliers, comme les émoticônes[6], les acronymes, les fautes (orthographiques, typographiques, mots collés, etc.) d'une très grande morpho-variabilité et d'une créativité explosive. Ces phénomènes doivent leur origine au mode de communication (direct ou semi-direct), à la rapidité de composition du message ou aux contraintes technologiques de saisie imposées par le matériel (terminal mobile, téléphone, etc.). J'ai introduit le terme **phonécriture** ou **phonécrit** comme toute forme écrite qui utilise un type d'écriture phonétique sans contraintes ou avec des règles établies par l'usage[7]. Le traitement automatique des courriels est extrêmement difficile à cause de son caractère imprévisible (Beauregard, 2001; Kosseim et Lapalme, 2001; Kosseim et al., 2001; Cohen, 1996b) : des textes trop courts ($\approx$ 11 mots par courriel), régis par une syntaxe mal orthographiée et/ou pauvre. Ceci impose donc d'utiliser des outils de traitement automatique robustes et flexibles. On voulait proposer la combinaison des méthodes d'apprentissage supervisé et non supervisés afin d'effectuer le routage de courriels. Des approches probabilistes, fondées sur des mots et des n-grammes de lettres (Jalam et Chauchat, 2002; Miller et al., 2000) ont aussi été utilisées. La catégorisation thématique est au cœur de nombreuses applications du TAL. Ce contexte fait émerger un certain nombre de questions théoriques

[4] Extraits de la presse française sur l'Internet et « Les rêveries du promeneur solitaire » De J.J. Rousseau, disponible sur `http://abu.cnam.fr/BIB/auteurs`

[5] « Discours de la méthode » de R. Descartes, disponible à la même adresse que le précedent et « Discours sur l'origine et les fondements de l'inégalité parmi les hommes » de J.J. Rousseau, disponible à l'adresse `http://un2sg4.unige.ch/athena/rousseau/jjr_ineg.html`

[6] Symboles utilisés dans les messages pour exprimer les émotions, exemple le sourire **:-)** ou la tristesse **:-(**

[7] Par exemple **kdo** à la place de cadeau, **10ko** pour dictionnaire, **A+** à plus tard, **@2m1** à demain, etc.

nouvelles, en particulier en relation avec la problématique du traitement d'informations textuelles incomplètes et/ou très bruitées (Kosseim et al., 2001).

L'ensemble de résultats de ces études, faisant partie d'abord du DEA de Rémy Kessler, puis de sa thèse, a été publié dans la revue *Ingénierie des Systèmes d'Information* (Kessler et al., 2006) et dans les mémoires du congrès VSST'04 (Kessler et al., 2004b) à Toulouse et dans MICAI'07 (Kessler et al., 2007) à Aguascalientes (Mexique).

Position du problème et prétraitement

Soit une boîte aux lettres qui reçoit un grand nombre de courriels correspondant à plusieurs thématiques. Une personne doit lire ces courriers et les rediriger vers le service concerné (les courriels de problèmes techniques vers le service technique, ceux pour le service après vente seront redirigés en conséquence, etc.). Il s'agit donc d'automatiser cette tâche. Il existe des approches pour effectuer le traitement automatique de courriels en anglais (Kiritchenko et Matwin, 2001; Kosseim et al., 2001; Cohen, 1996a; Jake D. Brutlag, 2000). Cependant, il s'est avéré difficile de trouver des travaux sur les courriels en français (des corpus en anglais existent cependant pour la classification de *spams*). Pour générer le corpus on a créé une adresse électronique et on l'a abonné à plusieurs listes de diffusion : Football, jeux de rôles, ornithologie, cinéma, jeux vidéo, poèmes, humour... ainsi qu'à des *newsletters* : Sécurité informatique, journaux, matériel informatique... Il faut noter que l'évaluation du système s'effectue en fonction de la liste de diffusion émettrice.

La catégorisation de documents consiste à attribuer une classe à un document donné. Cette tâche peut-être vue comme une tâche de classification (Lewis et Ringuette, 1994), où l'on fait correspondre des objets à une classe définie par une fonction d'appartenance. Le calcul de cette fonction inconnue peut être formulé comme un problème d'apprentissage supervisé, non supervisé ou semi-supervisé. Les méthodes retenues reposent sur la représentation vectorielle.

La première partie du pré-traitement identifie les courriels et sépare le corps du message des pièces jointes (celles-ci, contenant des graphiques, sons, etc. dans une grande diversité de formats, posent un problème complexe). Un second processus supprime automatiquement la micro-publicité qui n'apporte aucune information mais, au contraire, ajoute du bruit risquant de gêner la catégorisation. Il s'agit en général (à plus de 95 % de cas) de publicités rajoutées au bas des courriels par les fournisseurs de service de messagerie électronique. Nous avons supprimé la micro-publicité à l'aide de recherche de patrons. Un dictionnaire a été constitué à partir de sites[8] décrivant les divers termes de phonécriture, afin de trouver leur équivalent en français. Cette « traduction » est réalisée avant la suppression de la ponctuation, car beaucoup de termes phonécrits sont composés à l'aide de ponctuation. Concernant la lemmatisation, dans un premier temps, nous avons effectué le pré-traitement avec un dictionnaire accentué. L'usage des caractères accentués n'étant pas systématique dans les courriels, on a

[8] `http://www.mobimelpro.com/portail/fr/my/dictionnaire_sms.asp` `http://www.mobilou.org/10kosms.htm` ou `http://www.affection.org/chat/dico.html`

adapté en conséquence le processus en utilisant un dictionnaire avec et sans accents. Le pré-traitement ayant réduit la taille du corpus ≈ 70 % (sur un corpus d'environ 950 000 occurences au départ, nous obtenons un corpus de $\approx 250\,000$ occurences après filtrage). Additionellement, les performances en catégorisation ont été legèrement améliorés.

La vectorisation du corpus produit une matrice de P vecteurs (nombre de courriels) à N dimensions (taille du lexique). Chaque composante $(\sigma_1^\mu, \sigma_2^\mu, \cdots, \sigma_N^\mu)$ contient la fréquence du terme i dans un courriel μ. La matrice est divisée aléatoirement en sous-ensembles d'apprentissage et de test (ou généralisation), puis traitée par des méthodes d'apprentissage. On a décidé d'utiliser l'algorithme *fuzzy k-means* (Bezdek, 1981; deGruijter et McBratney, 1988) pour l'apprentissage non supervisé et les SVM (Joachims, 1998) pour l'apprentissage supervisé.

Apprentissage non supervisé

L'algorithme Fuzzy *k-means* (Bezdek, 1981; deGruijter et McBratney, 1988) permet d'obtenir un regroupement des éléments par une approche floue avec un certain degré d'appartenance où chaque élément peut appartenir à une ou plusieurs classes, à la différence de *k-means*, où chaque exemple appartient à une seule classe (partition dure). L'intérêt d'utiliser *fuzzy k-means* dans le cadre de la classification thématique de courriers électroniques (Kessler et al., 2004b,a, 2006) consiste à router un message vers un destinataire prioritaire (celui avec le degré d'appartenance le plus élevé) et en copie conforme (Cc) ou cachée (Bcc) vers celui (ou ceux) dont le degré d'appartenance dépasse un certain seuil empirique établi à l'avance. Nous avons fait une implantation de l'algorithme avec la distance de Manhattan, cependant les résultats étant décevants, nous avons utilisé la distance euclidienne.

Initialisation aléatoire ou semi-supervisée ? *k-means* et *fuzzy k-means* sont des algorithmes performants mais fortement dépendants de l'initialisation (Fred et Jain, 2003). Nous étions donc confrontés au problème de l'initialisation des centroïdes. Nous avons d'abord testé la méthode avec des initialisations aléatoires, mais l'erreur d'apprentissage, ϵ_a était ≈ 25 % dans le meilleur des cas (voir figure 2.2). De même, l'erreur en généralisation ϵ_g était toujours assez importante. Ceci est dû au fait que l'algorithme semble piégé dans des minima locaux. Nous avons donc décidé d'initialiser, de façon semi-supervisée, en prenant un petit nuage d'exemples (avec leur classe) afin d'avoir des points de départ mieux situés pour les centroïdes. Sur la figure 2.2, sont illustrés les résultats que nous avons obtenus sur 10 ensembles d'apprentissage tirés au hasard. La comparaison entre l'initialisation aléatoire et celle semi-supervisée montre que cette dernière est nettement supérieure. L'initialisation semi-supervisée a donc résolu le problème. Il est cependant important de rappeler que l'apprentissage avec *k-means* est toujours non supervisé, et qu'il suffit d'initialiser avec un nombre d'exemples entre 10-20 % pour obtenir $\epsilon_g < 10$ %. Nous avons voulu par ailleurs, connaître l'incidence du paramètre de flou f afin d'améliorer les résultats. Nous avons donc effectué une série de tests en ne faisant varier que ce paramètre, f allant de 2 à 50. Les résultats de la figure 2.2 à droite montrent qu'au-delà d'une valeur de 10 (en échelle logarithmique), les variations sur ϵ_g sont négligeables. Nous avons retenu $f = 6$ comme

valeur finale.

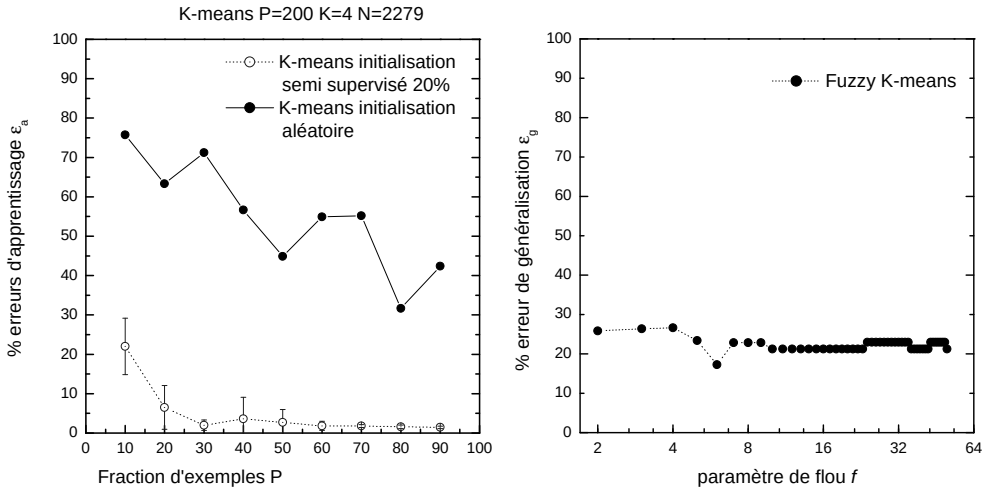


FIG. 2.2: *Comparaison entre l'initialisation aléatoire et l'initialisation semi-supervisée. Dans tous les cas, les moyennes sont calculées sur 10 tirages aléatoires. A droite : incidence du log_2(paramètre flou f) sur l'erreur de généralisation ϵ_g. $P = 160$ courriels en apprentissage et 40 en test*

N-grammes

Un n-gramme de lettres est une séquence de n caractères consécutifs. Pour un document donné, on peut générer l'ensemble des n-grammes ($n = 1, 2, 3, ...$) en déplaçant une fenêtre glissante de n cases sur le corpus. À chaque n-gramme, on associe une fréquence. De nombreux travaux (Damashek, 1995; Dunning, 1994) ont montré l'efficacité de l'approche des n-grammes comme méthode de représentation des textes pour des tâches de classification : soit la recherche d'une partition en groupe homogène, soit pour leur catégorisation (Jalam et Chauchat, 2002). Comparativement à d'autres techniques, les n-grammes de lettres capturent automatiquement les racines des mots (Grefenstette, 1996) et ils opèrent indépendamment des langues (Dunning, 1994), contrairement aux systèmes basés sur les mots. Ces méthodes sont tolérantes au bruit (fautes d'orthographe). (Miller et al., 2000) montre que des systèmes basés sur les n-grammes gardent leurs performances malgrè des taux de bruit de 30 %. Enfin ces techniques n'ont pas besoin d'éliminer les mots-outils ni de procéder à la racinisation (lemmatisation ou radicalisation), traitements qui augmentent la performance des techniques basées sur les mots. Par contre, pour les systèmes n-grammes, des études comme celle de (Sahami, 1999) ont montré que la performance ne s'améliore pas après l'élimination de mots fonctionnels et de racinisation.

Etant donné la faible quantité de mots ($\approx$ 11 mots) contenues dans ce corpus de courriels, des tests préliminaires concernant leur classification, ont montré que les n-grammes de lettres ont des performances supérieures à celles de n-grammes de mots. Pour tenir compte de ces observations, nous avons donc décidé d'utiliser les n-grammes de lettres à la place des n-grammes de mots comme termes. Nous effectuons pour cela un découpage des messages en n-grammes avec ou sans lem-

matisation afin de pouvoir effectuer un comparatif entre les deux méthodes ainsi qu'en fonction de la racinisation des termes.

Apprentissage supervisé

Les SVM ont déjà été appliquées au domaine de la classification du texte dans plusieurs travaux (Grilheres et al., 2004; Vinot et al., 2003; Joachims, 1998, 1999), mais toujours en utilisant des corpus bien rédigés (des articles journalistiques, scientifiques...). On a testé plusieurs implantations des machines SVM (Lia_SCT (Béchet et al., 2000), SVMTorch[9], Winsvm[10], M-SVM[11]) afin de sélectionner la plus efficace. Nous avons décidé d'utiliser SVMTorch (Collobert et Bengio, 2000), qui permet une approche multiclasse des problèmes de classification. Elle utilise le principe *One-against-the-Rest*, où chaque classe est comparée à l'ensemble des autres afin de trouver un hyperplan séparateur. Les résultats de la section 2.2.1 ont été obtenus avec une fonction à noyau simple, qui s'est averée la plus performante. Les erreurs d'apprentissage et de test sont plus importantes avec d'autres fonctions noyaux.

La méthode hybride

On a décidé de combiner les méthodes d'apprentissage supervisé et non supervisé afin de tirer parti des avantages de chacune d'entre elles. En effet, l'apprentissage non supervisé avec *k-means* donnait de résultats acceptables lors de la phase d'apprentissage mais faisait beaucoup d'erreurs en généralisation. D'un autre côté, l'apprentissage avec les SVM est supervisé et comme on le sait, il est très coûteux d'avoir de grands ensembles de données etiquetées. Nous effectuons un tirage aléatoire afin de constituer les matrices d'apprentissage γ_1 et de test γ_2. Nous effectuons ensuite un apprentissage non supervisé avec *k-means* sur la matrice γ_1 qui fournit la classe prédite pour chaque courriel. La deuxième étape consiste à présenter γ_1 à la machine à vecteurs de support, celle-ci pouvant dès lors effectuer un apprentissage supervisé à l'aide des étiquettes fournies par *k-means*. La généralisation est effectuée par la méthode SVM sur l'ensemble γ_2 à partir des vecteurs de support trouvés précédemment. Ainsi, plusieurs tests statistiquement indépendants ont été effectués.

2.2.1 Expériences et discussion

Les corpus de $P = \{200, 500, 1\,000, 2\,000\}$ courriels ont été générés sur $k = 4$ classes {**football, jeux de rôles, cinéma, ornithologie**}. Les tests ont été effectués à 50 ou 100 tirages aléatoires, dans le cas de n-grammes ou de mots, respectivement. La figure 2.2 présente à gauche une comparaison entre une initialisation aléatoire et une initialisation semi-supervisée.

La figure 2.3 présente les résultats obtenus sur un corpus de $P = 500$ courriels. À gauche, on montre l'erreur d'apprentissage ϵ_a avec une initialisation semi-supervisée

[9] http://www.idiap.ch/
[10] http://liama.ia.ac.cn/PersonalPage/lbchen/winsvm.htm
[11] http://www.loria.fr/~guermeur/

pour *k-means*. À droite, l'erreur de généralisation ϵ_g de SVM avec un apprentissage supervisé. Bien sur, l'erreur ϵ_g est faible : inférieur à 10 % au delà de 50 % des exemples appris. Cela correspond à la meilleure situation possible en apprentissage, mais les SVM supervisées on besoin de données étiquettées dans l'ensemble d'apprentissage, ce qui n'est pas toujours disponible.

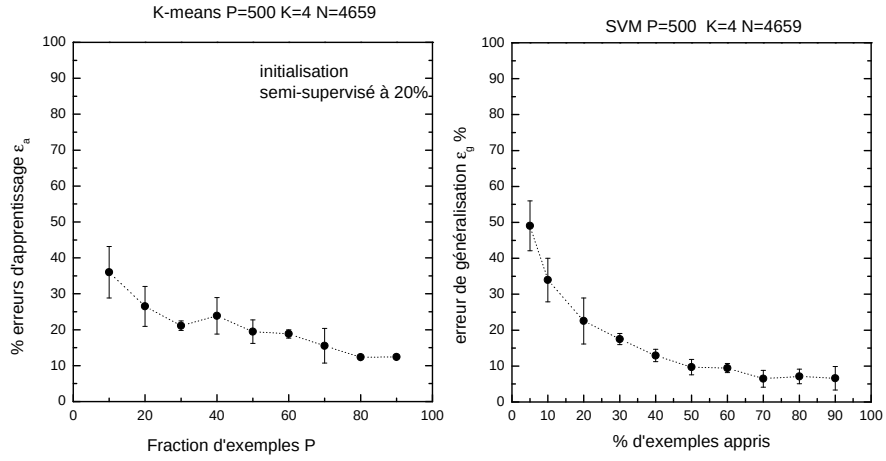


FIG. 2.3: *Erreur d'apprentissage ϵ_a pour k-means à gauche et de généralisation ϵ_g SVM à droite, $P = 500$ courriels. $K = 4$ classes, N = dimension des matrices.*

La figure 2.4 présente les résultats obtenus sur un corpus de $P = 500$ courriels et un découpage en n-grammes de lettres, avec et sans lemmatisation. Nous avons effectué des tests avec $n = \{2, 3, 5\}$ et nous avons trouvé que les bigrammes produisent les meilleurs résultats. Nous montrons l'erreur de généralisation ϵ_g pour la méthode hybride.

Les figures 2.5 comparent les résultats de la méthode hybride et des SVM sur des corpus de P={200, 500, 1 000, 2 000} courriels. Dans le cas hybride, nous avons combiné un apprentissage non supervisé par *k-means* (initialisation semi-supervisée de $0,05P$ ou $0,20P$ courriels) et supervisé pour SVM. Nous constatons que la performance ne se détériore pas en augmentant la taille du corpus. On voit aussi que les performances de généralisation de la méthode hybride sont très proches de celles des SVM. Notons que dans toutes les courbes des figures 2.5 l'unité de base est un unigramme de mot.

On a analysé les messages mal catégorisés afin de comprendre pourquoi le système les a mal classés. Une analyse *a posteriori* a montré que les messages trop courts, présentant des termes communs à plusieurs thématiques ou une combinaison de ces deux caractéristiques sont souvent mal catégorisés. Nous présentons deux exemples de messages qui n'ont pas été bien classés par notre système.

Catégorie **Jeux de rôles**, classé **Sport** :

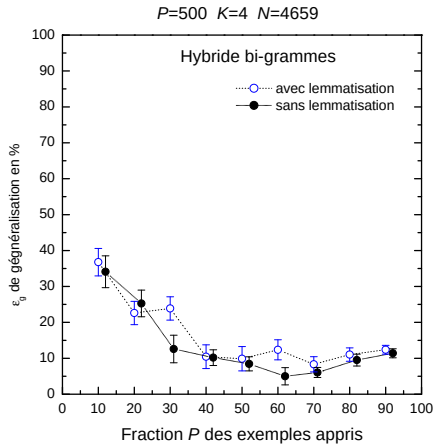


FIG. 2.4: *Erreur de généralisation pour la méthode hybride avec bi-grammes de lettres (50 tirages aléatoires) sur un corpus de P = 500 courriels. Le fait de ne pas lemmatiser améliore sensiblement les résultats. K = 4 classes, N dimension des matrices.*

```
From - Sun May 02 14 :40 :16 2004
Received : from [266.28.166.929] by n2.grp.scd.yahoo.com
To : <Shadowrun-france@yahoogroupes.fr>
In-Reply-To : <c70ti1+c4d8@eGroups.com>
From : Valerie <val@unicom.net>
Mailing-List : list <Shadowrun-france@yahoogroupes.fr>;
Date : Sun, 2 May 2004 05 :24 :15 -0700 (PDT)
Subject : [Shadowrun-france] Re : bonsoir_le_groupe
t'as raison... c'est pas brillant... au secours!!!!!!!!!!!!!!!!
belle soirée Valérie
```

Catégorie **Sport**, classé **Cinéma** :

```
From - Thu Mar 18 12 :49 :15 2004
Received : (qmail 64796 invoked from network); 18 Mar 2004 11 :32 :10
-0000
To : france-foot@yahoogroupes.fr
From : Jean LE COUTEAUX <jean@net.com>
Mailing-List : list france-foot@yahoogroupes.fr;
Date : Thu, 18 Mar 2004 12 :33 :14 +0100
Subject : [france-foot] CDF - 1/4
Nantes (L1) 1-1 Rennes (L1)
Ôh purée, j'y etais et une fois de plus j'ai été décu.. On aurait dit
un mauvais film avec de mauvais acteurs.. C'était mou, il n'y avait
pas d'action... Le scénario était couru d'avance, chacun en défense
et on attend la fin...
```

Le premier est trop court et trop vague pour permettre une catégorisation adéquate. Le second appartient à la catégorie **Sport** mais il comporte plusieurs termes (**mauvais_film**, **mauvais_scénario**, **action**, **la_fin**) de la catégorie **Cinéma**. Ceci illustre une partie des difficultés inhérentes à cette tâche, qui n'est pas évidente même pour les humains.

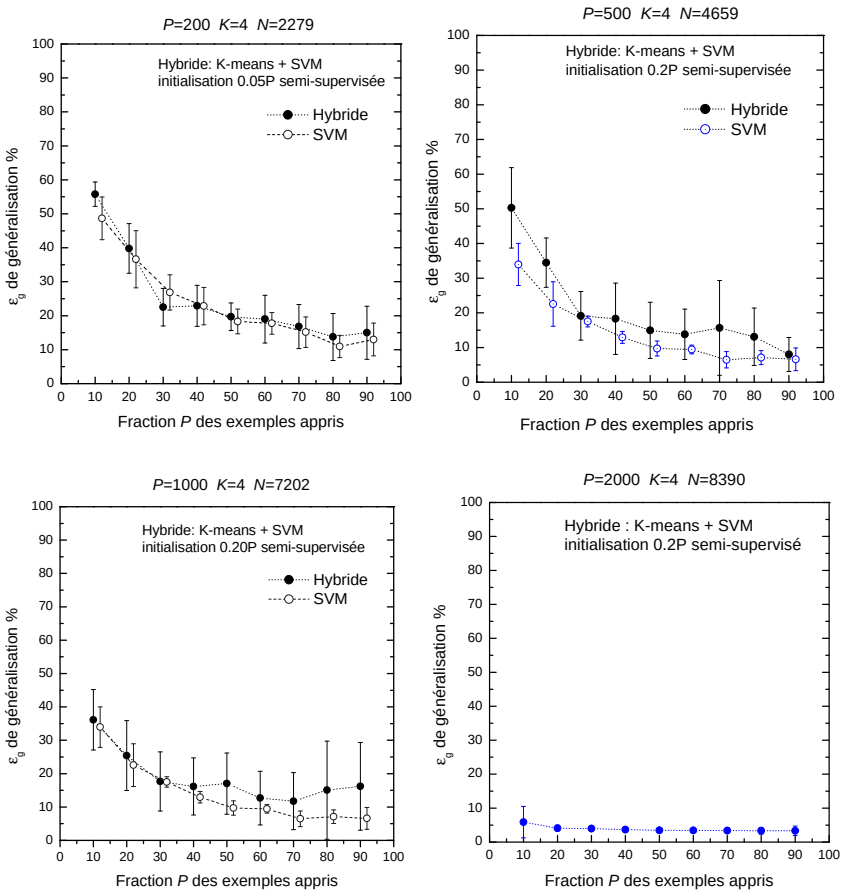


FIG. 2.5: *Méthode hybride vs. SVM. K = 4 classes, N = dimension des matrices, 100 tirages aléatoires. P = 200 à gauche et 500 courriels à droite. En bas à gauche P = 1 000, à droite 2 000 courriels. L'erreur est négligeable au délà de 20 % des exemples appris. Par souci de clarté seulement la courbe de la méthode hybride est presentée, celle-ci se confondant avec la courbe SVM.*

2.3 E-Gen : traitement automatique des offres d'emploi

La croissance exponentielle de l'Internet a permis le dévéloppement des sites d'offres d'emploi en ligne (Bizer et al., 2005; Rafter et al., 2000a; Rafter et Smyth, 2001). Cependant, les réponses des candidats représentent une grande quantité d'information qui ne peut pas être gérée efficacement par les entreprises (Bourse et al., 2004; Morin et al., 2004; Rafter et al., 2000b). En conséquence, il est nécessaire de la traiter d'une manière automatique ou assistée. Le LIA et Aktor Interactive[12], agence de communication française spécialisée dans l'*e-recruiting*, développent le système E-Gen pour résoudre ce problème. L'Agence Nationale de la Recherche Technologique (ANRT)[13] et Aktor Interactive, ont financé le travail de thèse de Rémy Kessler (contrat CIFRE numéro 172/2005).

[12] http://www.aktor.fr
[13] http://www.anrt.fr

Le système E-Gen se compose de deux modules principaux :
- Un module d'extraction de l'information à partir de corpus des courriels provenants de listages des offres d'emploi.
- Un module pour analyser et calculer un classement de pertinence du profil du candidat (lettre de motivation et curriculum vitæ).

Afin d'extraire l'information utile, ce premier module analyse le contenu des courriels d'offres d'emploi. Cette étape présente des problèmes intéressants liés au TAL : les textes des offres sont écrits dans un format libre, sans structure, avec certaines ambiguïtés, des erreurs typographiques, etc. Une des principales activités de l'entreprise est la publication d'offres d'emploi sur les sites d'emploi en ligne pour les sociétés ayant un besoin en recrutement. Face à la grande quantité d'information disponible sur internet et aux nombres importants de jobboards (spécialisés[14], non spécialisés[15] ou locaux[16]), Aktor a besoin d'un système capable de traiter rapidement et efficacement celles-ci afin de pouvoir par la suite les diffuser. Pour cela, Aktor utilise un système automatique pour envoyer les offres d'emploi en format XML (*Robopost Gateway*). Au cours de cette première étape, il est donc nécessaire d'identifier les différentes parties de l'offre d'emploi et de plus d'extraire certaines informations pertinentes (contrat, salaire, localisation, etc.). Auparavant, cette première étape était une tâche manuelle : on demande aux utilisateurs de copier et coller les offres d'emploi dans le système d'information de l'entreprise. Ces travaux présentent seulement le premier module du système E-Gen et sa performance sur la tâche d'extraction et de catégorisation.

2.3.1 Vue d'ensemble du système

Nous souhaitions développer un système qui répond aussi rapidement et judicieusement que possible au besoin de la société Aktor, et donc aux contraintes du marché de recrutement en ligne. Dans ce but, une boîte eléctronique reçoit les courriels (parfois avec un fichier attaché) contenant les offres d'emploi. Après l'identification de la langue, E-Gen analyse le message. Le texte contenant l'offre d'emploi est extrait à partir du fichier attaché. Un module externe, wvWare traite le document MS-Word et produit une version texte du document decoupé en segments[17].

Après un filtrage et une lemmatisation, on peut utiliser la représentation vectorielle afin d'attribuer une étiquette avec SVM. Par la suite, cette séquence d'étiquettes est traitée par un processus correctif qui la valide ou qui propose une meilleure séquence. À la fin du traitement, un fichier XML est généré et envoyé au système d'information d'Aktor.

Lors de la publication de l'offre d'emploi, un certain nombre d'informations est requis par l'entreprise. Ainsi il faut trouver ces champs dans l'annonce de l'offre afin de les incorporer dans le format XML. Nous avons mis au point différentes solutions à

[14] `http://www.admincompta.fr` (comptabilité), `http://www.lesjeudis.com` (informatique)
[15] `http://www.monster.fr`, `http://www.cadremploi.fr`, `http://www.cadresonline.com`
[16] `http://www.emploiregions.com`, `http://www.regionsjob.com`
[17] `http://wvware.sourceforge.net`. La segmentation de textes MS-Word étant un vrai cassetête, on a opté par un outil existant. Dans la majorité des cas, il sectionne en paragraphes le document.

base de règles afin de localiser des informations telles que salaires, lieu de travail, noms d'entreprises, contrat, référence, durée de la mission, etc.

2.3.2 Corpus et modélisation

Un sous-ensemble de données a donc été sélectionné à partir de la base de données d'Aktor. Ce corpus regroupe plusieurs sortes de listes d'emploi en différentes langues, mais l'étude porte sur les offres en français (le marché français représente l'activité principale d'Aktor). Ce sous-ensemble a été nommé *Corpus de référence*. Un exemple d'offre d'emploi est présenté dans la table 2.1. L'extraction à partir de la base de données

```
Ce groupe français spécialisé dans la prestation d'analyses chimiques,
recherche un :
RESPONSABLE DE TRANSFERT LABORATOIRE. Sud Est.
En charge du regroupement et du transfert d'activités de différents
laboratoires d'analyses, vous etudiez, conduisez et mettez en oeuvre
le sequencement de toutes les phases necessaires à la realisation de ce
projet, dans le respect du budget previsionnel et des delais fixes.
Vos solutions integrent l'ensemble des parametres de la demarche (social,
logistique, infrastructures et materiels, informatique) et dessinent le
fonctionnement du futur ensemble (Production, methodes et accreditations,
developpement produit, commercial).
De formation superieure Ecole d'ingenieur Chimiste (CPE) option chimie
analytique environnementale, vous avez deja conduit un projet de
transfert d'activite.
La pratique de la langue anglaise est souhaitee. Merci d'adresser
votre candidature sous la reference VA 11/06 par e-mail
beatrice.lardon@atalan.fr
```

TAB. 2.1: *Exemple d'offre d'emploi.*

d'Aktor a permis d'avoir un corpus de taille importante, sans catégorisation manuelle. Une première analyse a montré que les offres d'emploi se composent souvent de blocs d'information semblable qui demeurent, cependant, fortement non structurées. Une offre d'emploi est composée de quatre blocs :

1. TITRE : titre probable de l'emploi ;
2. DESCRIPTION : bref résumé de l'entreprise qui recrute ;
3. MISSION : courte description de l'emploi ;
4. PROFIL : qualifications et connaissances exigés pour le poste. Les contacts sont généralement inclus dans cette partie.

L'automate de Markov.

Les resultats préliminaires ont montré que la catégorisation de segments sans utiliser leur position dans l'offre d'emploi peut être une source d'erreurs. Nous avons constaté que les SVM produisent globalement une bonne classification des segments individuels, mais les offres d'emploi sont rarement complètement bien

classées. En raison du grand nombre de cas, les règles ne semblent pas être la meilleure façon de résoudre le problème. Nous avons donc opté pour un automate de Markov à six états : DÉBUT (S), TITRE (1), DESCRIPTION (2), MISSION (3), PROFIL (4), FIN (E). On représente une offre d'emploi comme une succession d'états dans cette machine. Le corpus de référence a été analysé pour déterminer les probabilités de transition entre les états. La matrice M (2.2) montre ces probablities.

$$M = \begin{array}{c} \\ \text{DÉBUT} \\ \text{TITRE} \\ \text{DESCRIPTION} \\ \text{MISSION} \\ \text{PROFIL} \\ \text{FIN} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{DÉBUT} & \text{TITRE} & \text{DESCRIPTION} & \text{MISSION} & \text{PROFIL} & \text{FIN} \end{array} \\ \begin{pmatrix} 0 & 0,01 & 0,99 & 0 & 0 & 0 \\ 0 & 0,05 & 0,02 & 0,94 & 0 & 0 \\ 0 & 0,35 & 0,64 & 0,01 & 0 & 0 \\ 0 & 0 & 0 & 0,76 & 0,24 & 0 \\ 0 & 0 & 0 & 0 & 0,82 & 0,18 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{2.2}$$

Lobservation de cette matrice M nous renseigne sur la structure des segments d'une offre d'emploi. Celle-ci a une probabilité $p = 0,99$ de commencer par le segment DESCRIPTION mais il est impossible de commencer par MISSION ou PROFIL. De la même manière, un segment MISSION peut seulement être suivi par d'un segment MISSION ou par un segment PROFIL. Ceci nous a permis d'en deduire l'automate montré sur la figure 2.6.

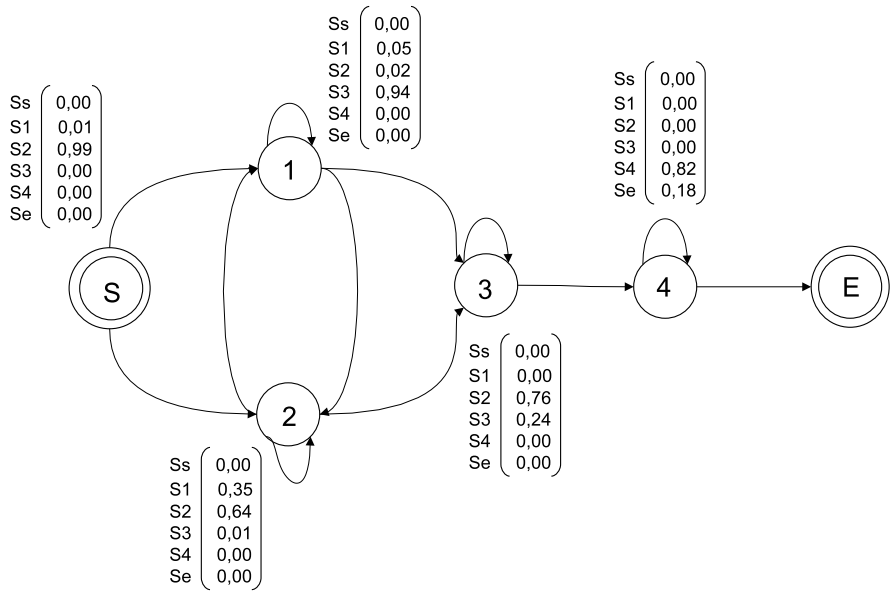


FIG. 2.6: *Machine de Markov à six états S_i, utilisée pour corriger les étiquettes erronées. i : S=DÉBUT, 1=TITRE, 2=DESCRIPTION, 3=MISSION, 4=PROFIL, E=FIN.*

Classification par SVM et un processus correctif.

Nous avons choisi les SVM pour cette tâche car nous avions déjà obtenus de bons résultats lors de travaux précédents sur la classification de courriels (c.f. Section 2.2). Nous utilisons l'implémentation Libsvm (Fan et al., 2005) qui permet de traiter les problèmes multiclasses de grandes dimensions. Les résultats obtenus par

SVM montrent une classification performante des segments. Pourtant, pendant la classification des offres d'emploi, quelques segments ont été classés incorrectement, sans un comportement régulier (un segment DESCRIPTION a été détecté au milieu d'un PROFIL, le dernier segment de l'offre d'emploi a été identifié comme TITRE, etc.). Afin d'éviter ce genre d'erreurs, on a appliqué un post-traitement basé sur l'algorithme de Viterbi (Manning et Schütze, 1999; Viterbi, 1967). La classification par SVM donne à chaque segment une classe afin de caractériser une offre en entier. Par exemple, pour la séquence S↦→2↦→2↦→1↦→3↦→3↦→4↦→E[18], l'algorithme classique de Viterbi calculera la probabilité de la séquence. Si la séquence est improbable, Viterbi renvoie 0. Si la séquence a une probabilité nulle le processus correctif renvoie la séquence avec une erreur minimale et une probabilité maximale (comparées à la séquence original produite par SVM). Les premiers résultats étaient intéressants, mais avec des temps de traitement assez grands. Nous avons introduit une amélioration en utilisant un algorithme *Branch and Bound* (Land et Doig, 1960) pour élaguer l'arbre : dès qu'une première solution est trouvée, son erreur et sa probabilité sont comparées chaque fois qu'une nouvelle séquence est traitée. Si la solution n'est pas meilleure, le reste de la séquence n'est pas calculée. L'utilisation de cet algorithme a permis d'obtenir la solution optimale, mais peut-être pas le meilleur temps (elle a un comportement exponentielle). Cependant cette stratégie calcule en $\approx$ 2 secondes les séquences de longueur $\leq$ 50 symboles.

2.3.3 Résultats et discussion

Un corpus de D=1 000 offres d'emploi avec P=15 621 segments a été utilisé. Chaque test a été effectué 20 fois avec une distribution aléatoire entre les corpus de test et d'apprentissage. La figure 2.7 montre une comparaison entre les résultats obtenus par les *Support Vector Machines* et le processus correctif. Les courbes présentent le nombre de segments non reconnus en fonction de la taille du corpus d'apprentissage. À gauche, on présente les résultats des SVM seules (ligne pointillée) appliquées sur la tâche de classification des segments. Les résultats sont bons et prouvent que même avec une petite fraction de patrons d'apprentissage (20% du total), le classifieur SVM obtient un faible taux de patrons mal classés (< 10% d'erreur). Le processus correctif (ligne continue) donne toujours de meilleurs résultats que les SVM quel que soit la fraction d'exemples d'apprentissage. À titre de comparaison, une classification nommée *Baseline* avec la classe la plus probable (étiquette PROFIL avec $\approx$ 40% d'apparition sur le corpus) obtient 60% d'erreur calculée sur tous les segments. La figure 2.7 à droite, est une comparaison entre les résultats obtenus par chaque méthode mais selon les offres d'emploi non reconnues. On observe une considérable amélioration du nombre d'offres d'emploi identifiés avec le processus correctif. SVM obtient un minimum de $\approx$ 50% des offres d'emploi non reconnues, et le processus correctif en obtient 20%, donc une amélioration de plus du 50% sur le score de SVM.

Une analyse des offres d'emploi mal classées, montre que $\approx$ 10% d'entre elles contient

[18] DÉBUT↦ DESCRIPTION↦DESCRIPTION↦TITRE↦MISSION↦MISSION↦PROFIL↦FIN

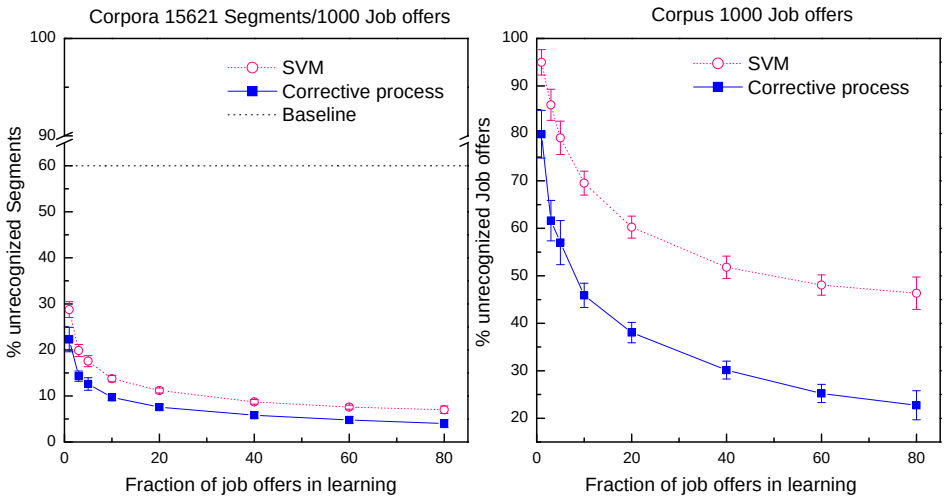


FIG. 2.7: *Resultats des SVM et de l'algorithme correctif par rapport aux segments à gauche, et aux offres d'emploi, à droite.*

une ou deux erreurs. Ces segments mal classés, correspondent généralement au bloc frontière entre deux catégories différentes (El-Bèze et al., 2007). Ainsi, la séquence obtenue de l'exemple 2.1 (cf. 2.3.2) est S↦→2↦→1↦→3↦→3 ↦→3↦→4↦→4↦→E et la séquence correcte est S↦→2↦→1↦→3↦→3↦→4↦→4↦→4↦→E. Le segment faux est reproduit ci bas :

> *De formation superieure Ecole d'ingenieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un* ***projet*** *de* ***transfert d'activite****.*

On voit que des termes importants présents dans deux catégories différentes amènent à une classification incorrecte. En particulier, des termes tels que **projet** et **transfert d'activité** correspondent aux catégories MISSION et PROFIL. Le segment est classé en tant que PROFIL. En fait, ce segment se trouve à la frontière entre les blocs MISSION | PROFIL et la séquence est probable (la probabilité de Viterbi n'est pas nulle), ainsi cette erreur n'est pas corrigée par le processus correctif. L'amélioration de la détection du bloc frontière est une des pistes que nous explorons actuellement (Reynar et Ratnaparkhi, 1997).

2.4 Conclusion et perspectives

J'ai presénté un panorama du modèle vectoriel de représentation de textes, qui malgré ses limitations, permet de traiter plusieurs tâches du TAL efficacement. Deux applications intéressantes, le routage de courriels et le traitement automatique des offres d'emploi ont ainsi été développés.

Le routage automatique de courriels est difficile en raison des particularités de cette forme de communication. Il s'agit d'une tâche où l'on travaille avec des événements rares. Nous avons effectué des processus de pré-traitement (filtrage, traduction, lemmatisation) afin de représenter les courriels dans un modèle vectoriel. L'apprentissage

supervisé permet de classer plus pécisement des nouveaux courriels mais demande un étiquetage préalable qui n'est pas toujours facile à mettre en œuvre. Par contre, bien que l'erreur d'apprentissage de *fuzzy k-means* avec initialisation aléatoire semble importante, nous l'avons diminuée avec une initialisation semi-supervisée (impliquant un faible nombre d'exemples), afin de nous passer de données étiquetées. La méthode hybride, qui combine les avantages de l'apprentissage non supervisé de *k-means* pour pré-étiqueter les données, et du supervisé avec SVM pour trouver les séparateurs optimaux, a donné des résultats très intéressants. Des modèles de découpage en mots et en n-grammes de lettres ont été développés et testées. Nous avons confirmé que la non lemmatisation produit des meilleurs résultats. Ceci va dans le même sens que les conclusions tirées par (Sahami, 1999). Des combinaisons des n-grammes avec un lissage approprié, pourraient être envisagées. Nous nous sommes principalement intéressés à l'amélioration de l'apprentissage non supervisé, celui-ci ayant les résultats les plus bas au départ. Nos résultats montrent que la performance du système hybride est proche de celle des SVM à noyau linéaire. Une optimisation des SVM pourrait améliorer les performances du système. Ainsi, une implantation selon l'algorithme DDAG (Kijsirikul et Ussivakul, 2002) permettrait d'éliminer les régions erronées et d'obtenir une meilleure classification des nouvelles données.

Le traitement des offres d'emploi est une tâche aussi difficile car l'information y est toujours fortement non structurée. J'ai montré le module de catégorisation, premier composant d'E-Gen, système pour le traitement automatiquement des offres d'emploi. Les premiers résultats obtenus par les SVM étaient très intéressants ($\approx$ 10% d'erreur pour un corpus d'apprentissage de 80%), mais le processus correctif améliore ces résultats par $\approx$ 50% et diminue considérablement les erreurs du type segments isolés incorrectement classés, tout en restant dans des temps de calcul très raisonnables. Ce module d'E-Gen est actuellement en test sur le serveur d'Aktor et permet un gain de temps considérable dans le traitement quotidien des offres d'emploi. D'autres approches (récupération de l'information et résumé automatique) seront considérées pour résoudre le problème de correspondance des candidatures à une offre d'emploi, avec un coût minimal en termes d'intervention humaine. De même, une combinaison de plusieurs classifieurs (SVM, arbres, modèles probabilistes...) (Grilheres et al., 2004) pourrait améliorer les résultats des deux tâchées abordées. J'y reviendrai dans le chapitre 3 où il sera question de la classification d'opinions.

Chapitre 3

Détection automatique d'opinions

La phrase en bas est vraie.
La phrase du haut est fausse.

Je présenterai quelques modèles d'apprentissage numériques et probabilistes appliqués à la tâche de classification telle que définie dans le cadre du défi DEFT'07 (DÉfi Fouille de Textes) : la classification d'un texte suivant l'opinion qu'il exprime. Pour classer les documents, nous avons utilisé plusieurs classifieurs et une fusion. Une comparaison entre les résultats des données en validation et en test montrent une coïncidence remarquable, et mettent en évidence la robustesse et les performances de la fusion que nous proposons. Les résultats obtenus, en termes de précision, rappel et F-score sur les sous corpus de test sont très encourageants.

L'équipe du LIA, constituée par Marc El-Bèze, Frédéric Béchet, Nathalie Camelin et moi-même a présenté et publié les résultats de cette étude dans le congrès de l'Association Française pour l'Intelligence Artificielle[1] AFIA'07 à Grenoble (Torres-Moreno et al., 2007). Lors de cette compétition, notre équipe a remporté la première place du défi DEFT'07.

3.1 Contexte

Dans le cadre de la plate-forme AFIA'07, la 3ème édition du défi DEFT (Azé et Roche, 2005; Azé et al., 2006), a eu lieu. Ceci est la deuxième participation dans DEFT de l'équipe TALNE du LIA[2]. DEFT'07 a été motivé par le besoin de mettre en place des techniques de classification de textes suivant l'opinion qu'ils expriment. Concrètement, il s'agit de classer les textes de quatre corpus en langue française selon les opinions qui y sont formulées. La classification d'un corpus en classes pré-déterminées, et son corollaire

[1] `http://afia.lri.fr`

[2] Lors de la première competition, en 2005, notre équipe avait remporté le défi (El-Bèze et al., 2005). A l'époque, le problème était de classer les segments des allocutions de Jacques Chirac et François Mitterrand préalablement mélangées. Plus de détails concernant DEFT'05 seront exposés dans le chapitre suivant.

le profilage de textes, est une problématique importante du domaine de la fouille de textes. Le but d'une classification est d'attribuer une classe à un objet textuel donné, en fonction d'un profil qui sera explicité ou non suivant la méthode de classification utilisée. Les applications sont variées. Elles vont du filtrage de grands corpora (afin de faciliter la recherche d'information ou la veille scientifique et économique) à la classification par le genre de texte pour adapter les traitements linguistiques aux particularités d'un corpus. *A priori*, un travail de classification des avis paraît simple. Or, de nombreuses raisons font que le problème est complexe. Facteur aggravant : on ne dispose que de corpus de taille moyenne, déséquilibrés par rapport à leurs classes, qui risque de biaiser les algorithmes. La tâche proposée par DEFT'07 vise le domaine applicatif de la prise de décision. Attribuer une classe à un texte, c'est aussi lui attribuer une valeur qui peut servir de critère dans un processus de décision. Et en effet, la classification d'un texte suivant l'opinion qu'il exprime a des implications notamment en étude de marchés. Certaines entreprises veulent désormais pouvoir analyser automatiquement si l'image que leur renvoie la presse est plutôt positive ou plutôt négative. Des centaines de produits sont évalués sur Internet par des professionnels ou des internautes sur des sites dédiés : quel jugement conclusif peut tirer de cette masse d'informations un consommateur, ou bien encore l'entreprise qui fabrique ce produit ? En dehors du marketing, une autre application possible concerne les articles d'une encyclopédie collaborative sur Internet comme Wikipédia : un article propose-t-il un jugement favorable ou défavorable, ou est-il plutôt neutre suivant en cela un principe fondateur de cette encyclopédie libre ?

Les organisateurs du défi DEFT'07 ont mis à disposition quatre corpus hétérogènes :

aVoiraLire. 3 460 critiques et les notes qui leurs sont associées (beaucoup de sites de diffusion de critiques de films ou de livres[3] attribuent, en plus du commentaire, une note). Les organisateurs du défi ont retenu une échelle de 3 niveaux de notes. Ceci donne lieu à 3 classes bien discriminées : 0 (mauvais), 1 (moyen), et 2 (bien).

jeuxvideo. 4 231 critiques[4] avec une analyse des différents aspects du jeu – graphisme, jouabilité, durée, son, scénario, etc. – et une synthèse globale du jugement. Comme pour le corpus précédent, a été retenue une échelle de 3 niveaux de notes, qui donne les 3 classes 0 (mauvais), 1 (moyen), et 2 (bien).

relectures. 1 484 relectures d'articles scientifiques concernant les décisions des arbitres et renvoient des conseils et critiques aux auteurs. L'échelle comporte 3 niveaux de jugement. La classe 0 (rejet de l'article), la classe 1 (acceptation sous condition de modifications majeures ou en séance de posters) et la classe 2 (acceptations d'articles avec ou sous des modifications mineures). Ce corpus et le suivant ont subi un processus préalable d'anonymisation de noms des personnes.

debats. 28 832 interventions de députés portant sur des projets de lois examinés par l'Assemblée Nationale. À chaque intervention est associé le vote de l'intervenant sur la loi discutée, 0 (faveur) ou 1 (contre).

Le corpora ont été fournis scindés en deux parties : une partie des données ($\approx$ 60%) comme ensemble d'apprentissage et une autre partie ($\approx$ 40%) a été réservée pour les

[3] Voir par exemple `http://www.avoir-alire.com`

[4] Venant du site `http://www.jeuxvideo.com`

tests proprement dits. Sous peine de disqualification, aucune donnée, en dehors de celles fournies par le comité d'organisation ne pouvait être utilisée. Ceci exclut notamment l'accès aux sites Web ou à n'importe quelle autre source d'information. Le tableau 3.1 présente des statistiques brutes (nombre de textes et nombre de mots) des différents corpus. Des exemples portant sur la structure et les détails des corpus, peuvent être consultés dans le site du défi[5]. Intuitivement, la tâche de classer les avis d'opinion des articles scientifiques est la plus difficile des quatre car le corpus afférent contient beaucoup moins d'informations que les trois autres, mais d'autres caractéristiques particulières à chaque corpus ont aussi leur importance.

Corpus	Textes (A)	Mots (A)	Textes (T)	Mots (T)
aVoiraLire	2 074	490 805	1 386	319 788
jeuxvideo	2 537	1 866 828	1 694	1 223 220
relectures	881	132 083	603	90 979
debats	17 299	2 181 549	11 533	1 383 786

TAB. 3.1: *Statistiques brutes sur les quatre corpus d'apprentissage (A) et de test (T).*

Évaluation

Les algorithmes ont été évalués sur des corpus de test (T) avec des caractéristiques semblables à ceux d'apprentissage (A) (cf. tableau 3.1), en calculant le *F-score* des documents bien classés, moyenné sur tous les corpus :

$$F - score(\beta) = \frac{(\beta^2 + 1) \times \langle Précision \rangle \times \langle Rappel \rangle}{\beta^2 \times \langle Précision \rangle + \langle Rappel \rangle} \tag{3.1}$$

$$\langle Précision \rangle = \frac{\sum_{i=1}^{n} Précision_i}{n} \; ; \; \langle Rappel \rangle = \frac{\sum_{i=1}^{n} Rappel_i}{n} \tag{3.2}$$

Etant donné pour chaque classe i :

$$Précision_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents attribués à la classe } i\}} \tag{3.3}$$

$$Rappel_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents appartenant à la classe } i\}} \tag{3.4}$$

D'après les règles du défi, un document est attribué à la classe d'opinion i si :
- seule la classe i a été attribuée à ce document, sans indice de confiance spécifié ;
- la classe i a été attribuée à ce document avec un meilleur indice de confiance que les autres classes (s'il existe un indice de confiance).

[5] http://deft07.limsi.fr/corpus-desc.php

Un classifieur automatique peut attribuer à un document une distribution de probabilité sur les différentes classes au lieu de lui attribuer une seule classe. L'indice de confiance est la probabilité pour un document d'appartenir à une classe d'opinion donnée. Le *F*-score pondéré par l'indice de confiance a été utilisé, à titre indicatif, pour des comparaisons complémentaires entre les méthodes mises en place par les équipes.

3.2 Représentations de textes

Un même texte peut être représenté par les différents paramètres qu'il est possible d'en extraire. Les représentations les plus courantes sont les mots, *Parts Of Speech* (POS) ou lemmes. En nous inspirant de l'approche typique de l'analyse des opinions (Hatzivassiloglou et McKeown, 1997), nous utilisons un paramètre de représentation supplémentaire, une étiquette nommée *seed*. Un *seed* est un mot susceptible d'exprimer une polarité positive ou négative (Wilson et al., 2005). Notre protocole de construction du lexique de *seeds* consiste en deux étapes. Premièrement, une liste de mots polarisés est créée manuellement. Exemple : *aberrant, compliments, discourtois, embêtement,...* Afin de généraliser la liste de mots polarisés obtenue, chaque mot est remplacé par son lemme. Nous obtenons un premier lexique de 565 *seeds*. Deuxièmement, un modèle *BoosTexter* est appris sur les textes représentés en mots. Les mots sélectionnés par ce modèle sont filtrés manuellement, lemmatisés et ajoutés au lexique. Au final, nous obtenons un lexique d'environ 2 000 *seeds*. Une phrase représentée en *seeds* ne contient alors que les lemmes faisant partie de ce lexique.

Normalisation graphique. Le recours à une étape de pré-traitement comme la lemmatisation est motivé par le taux de flexion élevé de la langue française. Néanmoins, dans le problème qui nous occupe, il s'avère utile de ne pas voir disparaître nombre d'informations comme par exemple certains conditionnels ou subjonctifs. Dans une relecture d'article, la présence de propositions comme « *Il aurait été préférable* » ou « *il eût été préférable* » laisse supposer que l'arbitre n'est pas totalement en faveur de l'acceptation du texte qu'il a relu. Pour ne pas en être privés, nous avons bridé la lemmatisation pour un petit nombre de cas susceptibles de servir de points d'appui lors de la prise de décision. Pour au moins deux systèmes, les textes lemmatisés ont été soumis à une étape que l'on pourrait qualifier de normalisation graphique. Marc El-Bèze a developpé un système de génération de $\approx 30\,000$ règles qui ont permis de réunifier les variantes graphiques (essentiellement des noms propres) et de corriger un grand nombre de coquilles[6]. Il est à noter que certaines de ces fautes d'orthographe ont pu être introduites par l'étape de réaccentuation automatique que nous avons appliquée au préalable sur les quatre corpus.

[6]En cas d'ambiguïté, ces récritures sont faites en s'appuyant sur les contextes gauches ou droits (parfois les deux). Exemple : *Thé-Old-Republic* $\Rightarrow$ *the-Old-Republic*. Ces règles de réécriture avaient aussi pour but de combler certaines lacunes de notre lemmatiseur. Il n'est pas inutile de ramener à leur racine des flexions même peu fréquentes de verbes qui ne se trouvaient pas dans notre dictionnaire (comme *frustrer, gâcher*, ou *gonfler*). Enfin, quelques règles (peu nombreuses) avaient pour mission d'unifier sous une même graphie des variantes sémantiques.

Agglutination. Les différents exemples donnés ci-dessus font apparaître des regroupements sous la forme d'expressions plus ou moins figées. Celles-ci ont été constituées par application de règles régulières portant sur des couples de mots. Pour leur plus grande partie, les 30 000 règles que nous avons utilisées proviennent d'un simple calcul de collocation effectué selon la méthode du rapport de vraisemblance (Mani et Mayburi, 1999). Une autre part non négligeable est issue de listes d'expressions disponibles sur l'internet[7]. Nous y avons ajouté également des proverbes extraits de listes se trouvant sur des sites web[8].

3.3 Classifieurs

Les outils de classification de texte peuvent se différencier par la méthode de classification utilisée et par les éléments choisis afin de représenter l'information textuelle (mot, étiquette morpho-syntaxique –*Part Of Speech*, POS–, lemmes, stemmes, sac de mots, sac de *n*-grammes, longueur de phrase, etc.). Parce qu'il n'y a pas de méthode générique ayant donné la preuve de sa supériorité (dans tous les cas de classification d'information textuelle), nous avons décidé d'utiliser une combinaison de différents classifieurs et de différents éléments de texte. Cette approche nous permet, en outre, d'en déduire facilement les mesures de confiance sur les hypothèses produites lors de l'étiquetage. Neuf systèmes de décisions ont été utilisés avec les différents classifieurs présentés ci-bas et les différentes représentations présentées dans la section 3.2. Ainsi, il s'agit d'obtenir des *avis différents* sur l'étiquetage d'un texte. En outre, le but n'est pas d'optimiser le résultat de chaque classifieur indépendamment mais de les utiliser comme des outils dans leur paramétrage par défaut et d'approcher l'optimum pour la fusion de leurs résultats. Parce que ces outils sont basés sur des algorithmes de classification différents avec des formats d'entrée différents, ils n'utilisent pas les mêmes éléments d'information afin de caractériser un concept. Une combinaison de plusieurs classifieurs utilisant différentes sources d'information en entrée peut permettre d'obtenir des résultats plus fiables, évaluée par des mesures de confiance basées sur les scores donnés par les classifieurs. On fera une brève présentation des classifieurs utilisés.

LIA_SCT (Béchet et al., 2000) est un classifieur basé sur les arbres de décisions sémantiques (SCT-Semantic Classification Tree (Kuhn et De Mori, 1995)). Il suit le principe d'un arbre de décision : à chaque nœud de l'arbre une question est posée qui subdivise l'ensemble de classification dans les nœuds fils jusqu'à la répartition finale de tous les éléments dans les feuilles de l'arbre. La nouveauté des SCT réside dans la construction des questions qui se fait à partir d'un ensemble d'expressions régulières basées sur une séquence de composants. Leur ordre dans le vecteur d'entrée a donc une importance. De plus, chaque composant peut se définir suivant différents niveaux d'abstraction (mots et POS par exemple) et d'autres paramètres plus globaux peuvent également intégrer le vecteur (nombre de mots du document par exemple). Lorsque le SCT est construit, il prend des décisions

[7] Comme celle qui se trouve à l'adresse `http://www.linternaute.com/expression/recherche`
[8] `http://www.proverbes.free.fr/rechprov.php`

sur la base de règles de classification statistique apprises sur ces expressions régulières. Lorsqu'un texte est classé dans une feuille, il est alors associé aux hypothèses conceptuelles de cette feuille selon leur probabilité. LIA_SCT utilise les textes représentés en lemmes.

BoosTexter (Schapire et Singer, 2000) est un classifieur à large marge basé sur l'algorithme de *boosting* : *Adaboost* (Freund et Schapire, 1996). Le but de cet algorithme est d'améliorer la précision des règles de classification en combinant plusieurs hypothèses dites *faibles* ou peu précises. Une hypothèse faible est obtenue à chaque itération de l'algorithme de *boosting* qui travaille en re-pondérant de façon répétitive les exemples dans le jeu d'entraînement et en ré-exécutant l'algorithme d'apprentissage précisément sur ces données re-pondérées. Cela permet au système d'apprentissage faible de se concentrer sur les exemples les plus compliqués (ou problématiques). L'algorithme de *boosting* obtient ainsi un ensemble d'hypothèses faibles qui sont ensuite combinées en une seule règle de classification qui est un vote pondéré des hypothèses faibles et qui permet d'obtenir un score final pour chaque constituant de la liste des concepts. Les composants du vecteur d'entrée sont passés selon la technique du sac de mots et les éléments choisis par les classifieurs simples sont alors des n-grammes sur ces composants. Quatre de nos systèmes utilisent le classifieur *BoosTexter* :

- LIA_BOOST_BASELINE : la représentation d'un document se fait en mots. BoosTexter est appliqué en mode 3-grammes ;
- LIA_BOOST_BASESEED : un document est représenté en seeds, chaque seed est pondéré par son nombre d'occurrences en mode unigramme ;
- LIA_BOOST_SEED : un document est représenté par les mots et également par les seeds toujours pondérés par leur nombre d'occurrences en mode unigrammes ;
- LIA_BOOST_CHUNK : L'outil LIA_TAGG[9] est utilisé pour découper le document en un ensemble de syntagmes lemmatisés. Chaque syntagme contenant un *seed* ainsi que le syntagme précédent et suivant sont retenus comme représentation. Les autres syntagmes sont rejetés de la représentation du document. *BoosTexter* est appliqué en mode 3-grammes sur cette représentation.

SVMTorch (Collobert et al., 2002) est un classifieur basé sur les machines à support vectoriel (*SVM*) proposées par Vapnik (Vapnik, 1982, 1995). Dans nos expériences, la technique la plus simple du sac de mots est utilisée : un document est représenté comme un vecteur dont chaque composante correspond à une entrée du lexique de l'application et chaque composante a pour valeur le nombre d'occurrences de l'entrée lexicale correspondant dans le texte. Le système LIA_NATH_TORCH est obtenu avec *SVMTorch*. Le vecteur d'entrée est représenté par le lexique des *seeds*.

Timble (Daelemans et al., 2004) est un classifieur implémentant plusieurs techniques de *Memory-Based Learning* –*MBL*–. Ces techniques, descendantes directes de l'approche classique des k-plus-proches-voisins (*K Nearest Neighbor* k-*NN*) appliquée à la classification, ont prouvé leur efficacité dans un large nombre de

[9] http://www.lia.univ-avignon.fr/chercheurs/bechet/download_fred.html

tâches du traitement du langage naturel. Le paramétrage par défaut de *TiMBL* est un algorithme *MLB* qui construit une base de données d'*instances de base* lors de la phase d'entraînement. Comme pour SVM-Torch, une instance est un vecteur de taille fixe dont les composantes sont les entrées du lexique ayant pour valeur le nombre d'occurrences dans le document. À cela s'ajoute une composante indiquant quelle est la classe à associer à ce vecteur de paires {caractéristique-valeur}. Lorsque la base de données est construite, une nouvelle instance est classée par comparaison avec toutes les instances existantes dans la base, en calculant la distance de celle-ci par rapport à chaque instance en mémoire. Par défaut, *TiMBL* résout l'algorithme *1-NN* avec la métrique *Overlap Metric* qui compte simplement le nombre de composantes ayant une valeur différente dans chacun des 2 vecteurs comparés. Cette métrique est améliorée par l'*Information Gain* (Quinlan, 1986, 1993) qui mesure la pertinence des composantes du vecteur. Le système LIA_TIMBLE est formé de l'outil *TiMBL* appliqué sur les *seeds*.

Modélisation probabiliste uni-lemme (LIA_JUAN). Nous avons voulu simplifier au maximum un classifieur et savoir si les modèles n-grammes avec $n > 1$ apportent vraimment des éléments discriminants. Ce système met en place un classifieur incorporant des techniques élementaires sur les n-lemmes. Ces techniques, descendantes directes de l'approche probabiliste (Mani et Mayburi, 1999) appliquées à la classification de texte, ont prouvé leur efficacité dans le défi de 2005 (El-Bèze et al., 2005). Les textes ont été filtrés légèrement (afin de garder notamment des petites tournures comme la voix passive, les formes interrogatives ou exclamatives), un processus d'agregation de mots composés (via un dictionnaire simple), puis regroupés dans des mots de la même famille. Ce processus comporte une lemmatisation particulière. Ainsi, des mots tels que : *chantaient*, *chant*, *chantons*, et même *chanteurs* et *chanteuses* seront ramenés au lemme « chanter ». Nous avons limité notre modèle à $n = 1$, soit des uni-lemmes, ce qui nous évite de calculer beaucoup de coefficients de lissage. Nous avons transformé donc chaque document en un sac d'uni-lemmes. Puis nous avons calculé la classe d'appartenance d'un document comme :

$$P_t(w) \approx \prod_i \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{3.5}$$

Nous avons appliqué ce modèle d'uni-lemmes à tous les corpus, sans faire d'autres traitements particulières.

Modélisation probabiliste n-lemme (LIA_MARC). Nous avons envisagé ici de recourir à une modélisation somme toute classique en théorie de l'information, tout en cherchant à y intégrer quelques unes des spécificités du problème. La formulation que nous avons retenue initialement se rapproche de celle que nous avions employée lors du DEFT'05 (El-Bèze et al., 2005) :

$$\widetilde{t} = Arg_t \max P(t) \times P(w|t) = Arg_t \max P(t) \times P_t(w) \tag{3.6}$$

L'étiquette t pouvant prendre ses valeurs dans un ensemble de cardinal réduit à 2 ou 3 éléments [0-1] ou [0-2], *a priori* le problème pourrait paraître simple, et la quantité des données fournies suffisante pour bien apprendre les modèles. Même si le vocabulaire propre aux différents corpus n'est pas si grand (entre 9 000 mots

différents pour le plus petit corpus et 50 000 pour le plus grand), il reste que certaines entrées sont assez peu représentées. Aussi dans la lignée de ce qui se fait habituellement pour calculer la valeur du second terme de l'équation 3.6 nous avons opté pour un lissage de modèles n-lemmes (n allant de 0 à 3).

$$P_t(w) \approx \prod_i \lambda_3 P_t(w_i|w_{i-2}w_{i-1}) + \lambda_2 P_t(w_i|w_{i-1}) + \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{3.7}$$

L'originalité de la modélisation que nous nous sommes proposés d'employer dans le cadre de DEFT'07 réside essentiellement dans les aspects discriminants du modèle. Lors de l'apprentissage, les comptes des n-lemmes sont rééchelonnés en proportion de leur pouvoir discriminant. Ce dernier est estimé selon un point de vue complémentaire au critère d'impureté de Gini selon la formule suivante.

$$G(w,h) \approx \sum_i P_t^2(t|w,h) \tag{3.8}$$

Les entrées w et leurs contextes gauches h qui ne sont apparus qu'avec une étiquette donnée t et pas une autre, ont un pouvoir discriminant égal à 1. Ce critère a été lissé avec un sous-critère G' permettant de favoriser (certes dans une moindre mesure que G) les couples (w,h) qui n'apparaissent que dans 2 étiquettes sur 3. Notons tout d'abord que l'emploi de tels critères discriminants est une façon de pallier le fait que l'apprentissage par recherche d'un maximum de vraisemblance ne correspond pas vraiment aux données du problème. Deuxièmement, il est aisé de comprendre combien un regroupement massif des entrées lexicales par le biais des collocations (cf. section 3.2) peut avoir un effet déterminant sur le nombre des événements à coefficient discriminant élevé. Ces deux remarques visent à souligner que sur ce point particulier le fameux croisement entre méthode symbolique et numérique a son mot à dire. En dernier lieu, nous avons aussi adapté le calcul du premier terme $P(t)$ de l'équation 3.6 en combinant la fréquence relative de l'étiquette t avec la probabilité de cette même étiquette sachant la longueur du texte traité. Pour cela, nous avons eu recours à la loi Normale.

3.4 Résultats et discussion

Afin de tester nos méthodes et de règler leurs paramètres, nous avons decidé de scinder l'ensemble d'apprentissage (A) de chaque corpus en cinq sous-ensembles approximativement de la même taille (en nombre de textes à traiter). La procédure d'apprentissage a été la suivante : nous avons concaténé quatre des cinq sous-ensembles et gardé le cinquième pour le test. Les ensembles ainsi concaténés seront appelés dorénavant, ensembles de devéloppement (D) et le restant, ensemble de validation (V). Cinq expériences par corpus ont été ainsi effectuées tour à tour. Nous allons présenter nos résultats en deux parties : d'abord ceux obtenus sur les ensembles de devéloppement et validation où nous avons paramétré nos systèmes, et ensuite les résultats sur les données de test (T) en appliquant les algorithmes. On notera que les résultats different quelque peu des résultats officiels, car depuis la clôture du défi notre système à évolué.

3.4.1 Évaluation sur les corpus de validation

Le decoupage de chaque corpus en sous-ensembles de devéloppement (D) et de validation (V) est le fruit d'un tirage aléatoire. Ce découpage permet, selon nous, d'éviter de régler les algorithmes sur un seul ensemble d'apprentissage, ce qui pourrait conduire à deux travers, le biais expérimental et/ou le phenomène de surapprentissage. Sur la figure 3.1 et le tableau 3.2, nous montrons le *F*-score du système de fusion sur les quatre corpus (V). L'apprentissage a été fait sur les ensembles de devéloppement, et le *F*-score est calculé sur les cinq ensembles de validation (V). On peut constater que le corpus de relectures d'articles scientifiques est le plus difficile à traiter. En effet, ce corpus comporte le plus petit nombre de textes ($\approx$ 704 en devéloppement, $\approx$ 177 en validation). Il est aussi dur à classer étant donnée des particularités propres à ce corpus que nos avons detecté : les arbitres corrigent souvent le texte des articles à la volée (directement dans leurs commentaires), ce qui est une introduction de bruit. Nous y reviendrons lors de la discussion de nos résultats.

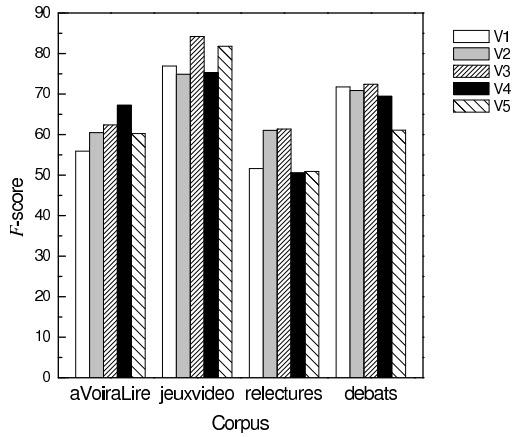


FIG. 3.1: *F-score obtenu par l'algorithme de fusion sur les cinq ensembles de validation (V). Les résultats sont regroupés par corpus.*

Corpus	Précision	Rappel	*F*-score	Corrects	Total
aVoiraLire (V)	0,6419	0,5678	0,6026	1 385	2 074
jeuxvideo (V)	0,8005	0,7730	0,7865	2 005	2 537
relectures (V)	0,5586	0,5452	0,5518	510	881
debats (V)	0,7265	0,7079	0,7171	12 761	17 299

TAB. 3.2: *Performances en Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les quatre corpus de validation (V).*

3.4.2 Évaluation sur les corpus de test

Nous avons défini l'ensemble d'apprentissage $\{A_j\} = \{D_j\} \cup \{V_j\}$; j ={**aVoiraLire**, **jeuxvideo**, **relectures**, **debats**}. Le tableau 3.3 montre les résultats de *F*-score pour l'en-

semble de test des quatre corpus.

Corpus	Précision	Rappel	*F*-score	Corrects	Total
aVoiraLire (T)	0,6540	0,5590	0,6028	931	1 386
jeuxvideo (T)	0,8114	0,7555	0,7824	1 333	1 694
relectures (T)	0,5689	0,5565	0,5626	353	603
debats (T)	0,7307	0,7096	0,7200	8 403	11 533

TAB. 3.3: *Performances en Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les quatre corpus de test (T).*

En figure 3.2, nous montrons les performances en *F*-score de chacun de nos classifieurs, ainsi que leurs moyennes sur les quatre ensembles de test. On constate que les classifieurs LIA_TIMBLE et LIA_SCT ont les perfomances les plus basses.

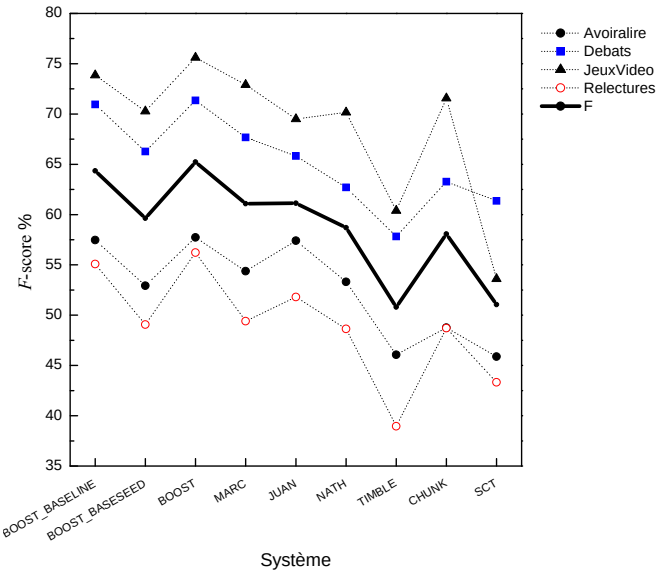


FIG. 3.2: *F-score de chacune des méthodes sur les quatre corpus de test.*

En figure 3.3, nous montrons les performances en *F*-score d'une fusion « incrémentale » des méthodes ajoutées. Cependant, l'ordre affiché n'a strictement aucun impact dans la fusion finale : il a été choisi uniquement pour mieux illustrer les résultats. On peut voir que nos résultats se placent bien au dessus de la moyenne des équipes participantes dans le défi DEFT'07, tous corpus confondus.

Sur la figure 3.4 nous montrons une comparaison du *F*-score de l'ensemble de validation (V) vs. celui de test (T), sur les quatre corpus. On peut constater la remarquable coïncidence entre les deux, ce qui signifie que notre stratégie d'apprentissage et de validation sur cinq sous-ensembles et de fusion de plusieurs classifieurs a bien fonctionné.

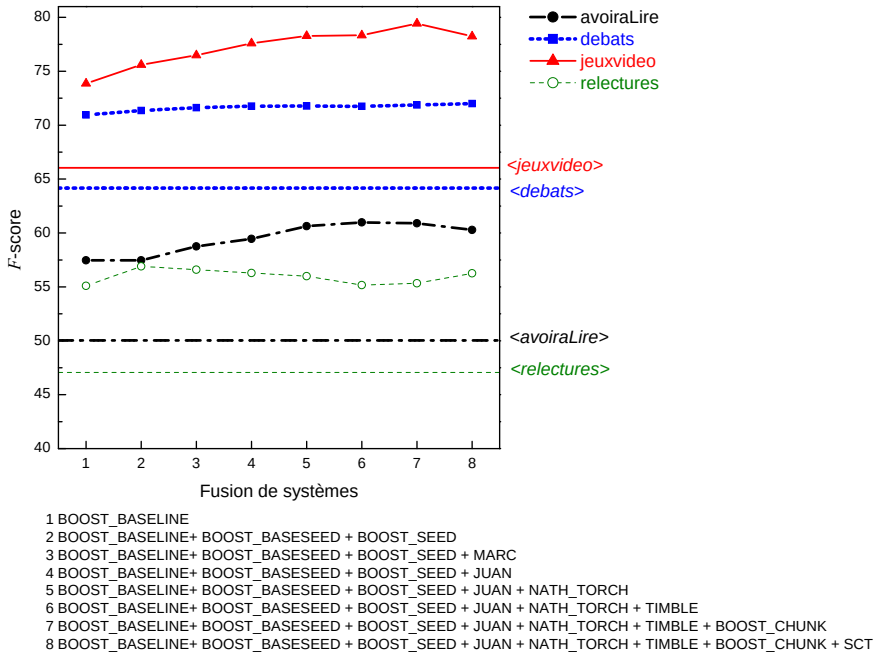


FIG. 3.3: *F-score de la fusion suivant nos 9 méthodes ajoutées. On affiche les moyennes des soumissions des participants : nous sommes au-dessus des moyennes, tous corpus confondus.*

3.4.3 Discussion

Nous avons constaté que l'utilisation de collocations et réécriture (cf. section 3.2) permet d'augmenter les perfomances des méthodes. Par exemple, avec la méthode probabiliste à base d'uni-lemmes sur le corpus de validation nous sommes passés de 1 285 à 1 310 bien classés (F=57,41 → 58,89) dans le corpus **aVoiraLire**, de 1801 à 1916 (F=70,77 → 75,15) en **jeuxvideo**, de 445 à 455 (F=48,36 → 49,53) en **relectures** et de 10 364 à 11 893 (F=62,21 → 67,12) en **debats**. Dans le corpus de test les gains sont aussi non négligeables. Nous sommes passés en **aVoirAlire** de 863 à 860 (F=57,40 → 56.32) ; de 7530 à 7635 (F=65,82 → 66,88) en **debats** ; de 1169 à 1205 (F=69,51 → 71,48) en **jeuxvideo** et de 317 à 313 (F=51,81 → 52,04) en **relectures**. Ceci confirme l'hypothèse que la réécriture aide à mieux capturer la polarité des avis.

Nous avons realisé une analyse *post-mortem* de nos résultats. Je présente quelques exemples de notices qui ont été mal classées par nos systèmes. J'ai delibérement gardé les notices dans leur état, même avec les fautes d'ortographe. Je vais montrer, en particulier, des avis venant du corpus de relectures d'articles scientifiques, corpus qui avait posé plus de difficultés aux algorithmes (F-score plus faible) que les autres. Pour la notice 3 :2 (**relectures**), l'article a été accepté mais notre système le rejette. Il comporte beaucoup d'expressions négatives comme : « *parties de l'article me paraissent déséquilibrées* », « *Le travail me paraît inachevé* », « *la nouvelle méthode proposée pose des problèmes complexes ... qui ne sont pas traités dans ce papier* », cependant il a été accepté.

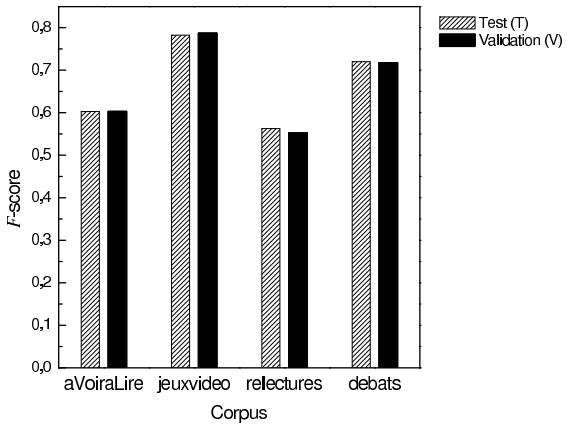


FIG. 3.4: *F-score de validation et de test pour chacun des corpus, obtenu par notre système de fusion.*

> ***Relectures 3 :2***
> ***Les différentes parties de l'article me paraissent déséquilibrées****. Les auteurs présentent d'abord un état de l'art dans le domaine de la visualisation des connaissances dans les systèmes de gestion de connaissances. Ils décrivent ensuite le serveur <anonyme /> et sa représentation des connaissances sous forme d'arbre en section 3 et une partie de la section 4. L'approche proposée par les auteurs (représentation par graphes n'est présentée qu'en 4.2 sur moins d'une page).* ***Les problèmes posés par cette méthode sont survolés par les auteurs****, ils font référence aux différents papiers traitant de ces problèmes et* ***n'exposent pas du tout*** *les heuristiques choisies dans leurs approches.* ***Le travail me paraît inachevé****, et* ***la nouvelle méthode proposée pose des problèmes complexes*** *au niveau de la construction de ce graphe* ***qui ne sont pas traités dans ce papier****.*

Considérez maintenant la notice 3 :36 (**relectures**) :

> ***Relectures 3 :36***
> *L'idée d'appliquer les méthodes de classification pour définir des classes homogènes de pages web est assez originale par contre, la méthodologie appliquée est classique. Je recommande donc un « weak accept » pour cet article.*

Nos systèmes l'ont classé 1 (accepté avec des modifications majeures), et après une lecture directe, on pourrait en déduire que la classe est 1, alors que la référence est 2 (accepté).

Notice 3 :6 (**relectures**). L'article a été accepté, alors que notre système le rejette. L'article arbitré est peut-être trop court, mais la relecture qui le concerne, elle l'est aussi :

> ***Relectures 3.6***
> *Article trop court pour pouvoir être jugé. Je suggère de le mettre en POster si cela est prévu.*

Pour la notice 3 :9 (**relectures**) on décèle le même problème : l'article est accepté alors que le système le rejette. Constatons que l'arbitre focalise uniquement sur des remarques de forme :

> ***Relectures 3 :9***
> *Question : comment est construit le réseau bayesien ? Un peu bref ici... Remarques de forme : page 2, 4ème ligne, « comprend » 5ème ligne : "annotées" ou "annoté" page 3 : revoir la phrase confuse précédant le tableau dernière ligne, répétition de "permet" page 5 : 7ème ligne accorder "diagnostiqué" et "visé" avec "états" ou avec "connaissances"*

Pour le texte de la notice 3 :567 (**relectures**), l'article en question est rejeté alors que le système l'accepte. Les phrases encourageantes au début finisent par être mitigées. Beaucoup d'expressions positives (« *bien organisé* », « *facile à suivre* », « *bibliographie est plutôt complète* », « *solution proposée et interessante et originale* », « *La soumission d'une nouvelle version ... sera intéressante* ») n'arrivent pas a renverser le rejet.

> ***Relectures 3 :567***
> *Commentaire : L'article est **plutôt bien organisé** (malgré de trop nombreux chapitres), le cheminement de **la logique est facile à suivre**. Cependant il y a de trop nombreuses fautes de français ainsi que d'anglais dans le résumé. **La bibliographie est plutôt complète**. **La solution proposée et interessante et originale**, cependant des notions semblent mal maîtrisées. Ainsi dans la section 8, la phrase «Cette convergence ne vient pas des algorithmes génétiques de manière intrinsèque, mais de l'astuce algorithmique visant à conserver systématiquement le meilleur individu dans la population » démontre une incompréhension du fonctionnement même d'un algorithme génétique. La soumission d'une nouvelle version modifiée de cet article, présentant également les premiers résultats obtenus avec le prototype à venir **sera intéressante pour la communauté**. Références : Originalité : Importance : Exactitude : Rédaction :*

Pour finir cette analyse, une notice du corpus de films, livres et spectacles, le texte 1 :10 du corpus **aVoiraLire**. Malgré des expressions telles que : « *...est un événement* », « *Agréable surprise* » ou encore « *l'image d'une cohérence artistique retrouvée* », la notice reste difficile à classer. Évidemment notre système se trompe. Je mettrai à mon tour le lecteur au défi de trouver la véritable classe[10].

[10] Si vous êtiez tenté de le mettre en classe 2 (bien), sachez que la classe véritable est la 1 (moyen).

aVoiraLire 1 :10
Depuis trente-six ans, chaque nouvelle production de David Bowie est un événement. Heathen , ne fait pas exception à cette règle. On reconnaît instantanément la patte de son vieux compère Tony Visconti. La voix de Bowie est mise en avant. Agréable surprise, surtout qu'elle n'a rien perdu depuis ses débuts. Là, commence le voyage. Ambiance, mélange dosé des instruments. Dès l'ouverture de l'album avec Sunday , un sentiment étrange nous envahit. Comme si Bowie venait de rentrer d'un voyage expérimental au coeur même de la musique. Retour aux sources. L'ensemble du disque est rythmé par cette pulsation dont le duo a le secret. Le tout saupoudré de quelques pincées d'électronique. Le groupe est réduit au minimum. Outre Bowie en chef d'orchestre et Visconti, David Torn ponctue les compositions de ses guitares aventureuses et Matt Chamberlain apporte de l'âme à la rythmique. Un quatuor à cordes fait une apparition, comme Pete Townshend (The Who) ou Dave Grohl (ex-batteur de Nirvana). Avec trois reprises réarrangées et neuf compositions originales, le 25e album de Bowie est à l'image d'une cohérence artistique retrouvée.

3.5 Conclusion et perspectives

La classification de textes en fonction des opinions qu'ils expriment reste une tâche très difficile, même pour les personnes. La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification par rapport à une autre. Nous avons décidé d'utiliser des approches de représentation numériques et probabilistes, afin de rester aussi indépendant que possible des sujets traités. Nos méthodes ont fait leurs preuves. Nous avons confirmé l'hypothèse que la réécriture (normalisation graphique) et les collocations aident à capturer le sens des avis. Ceci se traduit par un gain de performances. Nous avons présenté une stratégie de fusion assez simple de méthodes. Celle-ci s'est avérée robuste et performante. Nos F-scores sont au-dessus des moyennes sur les corpus de test, notamment sur celui de **jeuxvideo**. La stratégie de fusion a montré des résultats supérieurs à n'importe laquelle des méthodes individuelles. La dégradation en précision et rappel reste faible, même si nous n'avons pas écarté de la fusion des méthodes peut-être moins adaptées à cette problématique. La fusion est donc une façon robuste de combiner plusieurs classifieurs. Il faut souligner la remarquable équivalence entre les résultats obtenus lors de l'apprentissage et la prédiction sur les ensembles de test : à un point près de différence. Le corpus de relectures des articles scientifiques reste de loin le plus difficile à traiter. Nous avions déjà avancé l'hypothèse qu'en raison du faible nombre de notices, il serait difficile à classer. Il y a d'autres facteurs qui interviennent également. Les relectures comportent souvent des corrections adressées aux auteurs. Ceci vient bruiter nos classifieurs. Les relectures sont parfois trop courtes ou bien rédigées par des arbitres non francophones (encore une source de bruit) ou contiennent beaucoup d'anglicismes (*weak acceptation, boosting, support vector,...*). Un autre facteur, peut-être plus subtil : un article peut être lu par plusieurs arbitres (deux, trois voire plus) qui émettent des avis opposés. Dans une situation où les arbitres A et B acceptent l'article et un troisième C le refuse, normalement l'article doit

être accepté. Donc l'avis de C sera assimilé à la classe acceptée, et cela malgré son avis négatif. Enfin, le module de fusion n'a pas été optimisé, un même poids étant attribué au vote de chaque système. Ceci ouvre la voie à une possible amélioration.

Chapitre 4

Classification probabiliste du texte par leur contenu

–Ce soir vous n'êtes pas le président de la République,
nous sommes deux candidats à égalité (...),
vous me permettrez donc de vous appeler monsieur Mitterrand.
–Mais vous avez tout à fait raison, monsieur le Premier ministre.

Après avoir passé en revue la tâche classificatoire de DEFT'07, je présenterai dans ce chapitre une palette de modèles probabilistes employées dans le cadre de DEFT'05. La tâche proposée conjugue deux problématiques distinctes du TAL : l'identification de l'auteur (au sein de discours de Jacques Chirac, a pu être insérée une séquence de phrases de François Mitterrand) et la détection de ruptures thématiques (les thèmes abordés par les deux auteurs sont censés être différents). Pour identifier la paternité de ces séquences, nous avons utilisé des chaînes de Markov, des modèles bayésiens, et des procédures d'adaptation de ces modèles. Pour ce qui est des ruptures thématiques, nous avons appliqué une méthode probabiliste modélisant la cohérence interne des discours. Son ajout améliore les performances. Une comparaison avec diverses approches montre la supériorité d'une stratégie combinant apprentissage, cohérence et adaptation. Les résultats que nous obtenons, en termes de précision (0,890), rappel (0,955) et *F-score* (0,925) sur le sous-corpus de test sont très encourageants.

L'ensemble de résultats de cette étude[1], réalisée conjointement avec Marc El-Bèze et Fréderic Béchet, a été publié dans le congrès TALN'05/DEFT (El-Bèze et al., 2005) et dans la *Revue des Nouvelles Technologies de l'Information, RNTI 2007* (El-Bèze et al., 2007). Lors de cette compétition, notre équipe a remporté une fois de plus le défi DEFT.

[1]On remercie Éric Gaussier du Centre de Recherche Xerox Grenoble d'avoir mis à notre disposition un lexique de Noms Propres.

4.1 Introduction

L'atelier DEFT'05 (Azé et Roche, 2005) s'est deroulé dans le cadre des conférences TALN[2] et RECITAL[3] tenues en juin 2005 à Dourdan. Il a été motivé par le besoin de mettre en place des techniques de fouille de textes permettant soit d'identifier des phrases non pertinentes dans des textes, soit d'identifier des phrases particulièrement singulières dans des textes apparemment sans réel intérêt. Concrètement, il s'agissait de supprimer les phrases non pertinentes dans un corpus de discours politiques en français. Cette tâche est proche de la piste *Novelty* de TREC (Soboro, 2004) qui dans sa première partie consiste à identifier les phrases pertinentes puis, parmi celles-ci, les phrases nouvelles d'un corpus d'articles journalistiques. Pour mieux comprendre à quoi correspondait dans DEFT'05 la suppression des phrases non pertinentes d'un corpus de discours politiques (Alphonse et al., 2005), une brève description du but général[4] s'impose. Un corpus de textes, allocutions officielles issues de la Présidence (1995-2005) de Jacques Chirac a été fourni. Dans ce corpus, des passages issus d'un corpus d'allocutions (1981-1995) du Président François Mitterrand ont été insérés. Les passages d'allocutions de Mitterrand insérés sont composés d'au moins deux phrases successives et ils sont censés traiter une thématique différente[5]. Un corpus des discours de J. Chirac entrecoupés d'extraits de ceux de F. Mitterrand est ainsi constitué. Certaines informations sont supprimées de ce corpus afin de constituer les trois corpus ci-dessous :

- Corpus C1 : Aucune présence d'années ni de noms de personnes : ils ont été remplacés par les balises <DATE> et <NOM> ;
- Corpus C2 : Pas d'années : elles ont été remplacées par la balise <DATE> ;
- Corpus C3 : Présence des années et des noms de personnes.

Le but du défi consistait à déterminer les phrases issues du corpus de F. Mitterrand introduites dans le corpus composé d'allocutions de J. Chirac. Ce but est commun aux trois tâches T1, T2 et T3 relatives aux trois corpus C1, C2 et C3. Intuitivement, la tâche T1 est la plus difficile des trois car le corpus afférent C1 contient moins d'informations que les deux autres. Les résultats (calculés uniquement sur les phrases de F. Mitterrand extraites) peuvent être évalués sur un corpus de test (T) avec des caractéristiques semblables à celui de développement (D), en calculant le *F-score*. Dans le cadre de DEFT'05, le calcul du *F-score* a été effectué uniquement sur les phrases de Mitterrand, et il a été modifié (Azé et Roche, 2005) comme suit (cette réécriture suppose évidemment que $\beta = 1$ de façon à ne pas privilégier ni précision ni rappel) :

$$Fscore(\beta) = \frac{2 \times Nb\_phrases\_correctes\_extraites}{Nb\_total\_extraites + Nb\_total\_pertinentes} \tag{4.1}$$

- *Nb_phrases_correctes_extraites* : nombre de phrases qui appartiennent réellement au corpus de Mitterrand dans le fichier résultat ;

[2]Traitement Automatique des Langues Naturelles.

[3]Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues.

[4]Voir le site de DEFT'05 `http://www.lri.fr/ia/fdt/DEFT05` pour une description détaillée, les données et les résultats.

[5]Par exemple, dans les allocutions de Jacques Chirac évoquant la politique internationale, les phrases de François Mitterrand introduites sont issues de discours traitant de politique nationale.

– *Nb_total_extraites* : nombre de phrases données dans le fichier résultat ;
– *Nb_total_pertinentes* : nombre total de phrases de Mitterrand dans le corpus test.

On pourrait être tenté d'appliquer les méthodes employées habituellement en classification. *A priori*, un problème de classification à deux classes (ici Chirac et Mitterrand[6]) paraît simple. Or, de nombreuses raisons font que la question est complexe. Au terme d'une étude portant sur 68 interventions télévisées composées de 305 124 mots, (Labbé, 1990) distingue quatre périodes dans les discours de Mitterrand. L'une d'elles dénommée « Le président et le premier ministre » (octobre 1986 - mars 1988) n'est probablement pas la plus facile à traiter sous l'angle particulier proposé par le défi DEFT'05. Dans d'autres conditions, c'eût été loin d'être évident. Ici, on peut s'attendre à des difficultés accrues pour différencier deux orateurs qui se sont exprimés dans maints débats sur les mêmes sujets. Facteur aggravant : on ne dispose que d'un petit corpus déséquilibré. Pour la tâche T1, 109 279 mots pleins pour un président et 582 595 pour le second répartis dans 587 discours (dont la date n'est pas fournie). Pour donner une idée de la difficulté du défi, notons qu'une classification supervisée avec un perceptron optimal à recuit simulé (Minimerror présenté dans le chapitre 1) (Torres-Moreno et al., 2002) appliqué sur la catégorie grammaticale de mots (l'utilisation de tous les mots générant une matrice trop volumineuse) donne un taux d'extraction des segments Mitterrand décevant avec un *F-score* $\approx 0{,}43$; la méthode classique *K-means* sur les mêmes données, conduit à un *F-score* $\approx 0{,}40$. En comparaison, avec des classifieurs à large marge réputés performants tels que *AdaBoost* (Freund et Schapire, 1997) avec *BoosTexter* (Schapire et Singer, 2000) et *Support Vector Machines* avec SVM-Torch (Collobert et Bengio, 2001), on plafonne à un *F-score* $\approx 0{,}5$. Enfin, avec une méthode *baseline* vraiment simpliste, où on classerait tout segment comme appartenant à la catégorie *M*, on obtiendrait un *F-score* $\approx 0{,}23$ sur l'ensemble de développement et de 0,25 pour le test. Comme ces résultats se sont avérés décevants, nous avons décidé d'explorer des voies totalement différentes.

4.2 Modélisation

La chaîne de traitement que nous avons produit est constituée de quatre composants : un ensemble de modèles bayésiens (cf. 4.2.1), un automate de Markov (cf. 4.2.2), un modèle d'adaptation (cf. 4.2.3) et un réseau sémantique (cf. 4.2.4). Le seul composant totalement dédié à la tâche est l'automate. Le réseau sémantique dépend du domaine.

4.2.1 Modèles bayésiens

Guidée par une certaine intuition que nous aurions pu avoir des caractéristiques de la langue et du style de chacun des deux orateurs, une analyse des données d'apprentissage aurait pu nous pousser à retenir certaines de leurs caractéristiques plutôt que d'autres. En premier lieu, il aurait été naturel de tabler sur une caractérisation

[6]Nous désignons les deux derniers présidents par leur nom de famille, et pour plus de concision, il nous arrivera de remplacer « Mitterrand » et « Chirac » par les étiquettes *M* et *C*.

s'appuyant sur les différences de vocabulaire. Des études anciennes comme celles de (Cotteret et Moreau, 1969) sur le vocabulaire du Général de Gaulle, ou d'autres plus récentes (Labbé, 1990) partent du même présupposé. Pour plusieurs raisons, cette approche semble prometteuse mais comme on en rencontre tôt ou tard les limites, on est amené naturellement à ne pas s'en contenter. En effet, la couverture des thématiques abordées par les différents présidents est très large. Il est inévitable que les trajets politiques de deux présidents consécutifs se soient à maintes reprises recoupés. Par conséquent on observe de nombreux points communs dans leurs interventions. On suppose qu'à ces recouvrements viennent s'ajouter les reproductions conscientes ou inconscientes (citations ou effets de mimétisme).

Modélisation I, avec lemmatisation. Au travers d'une modélisation tout à fait classique (Manning et Schütze, 1999), nous avons testé quelques points d'appuis comme la longueur des phrases (LL), le pourcentage de conjonctions de subordination (Pcos), d'adverbes (Padv) ou d'adjectifs (Padj) et la longueur moyenne des mots pleins (Plm). Cinq de ces variables (Pcos, Padv, Padj, LL, et Plm) ont été modélisées par des gaussiennes p_i dont les paramètres ont été estimés sur le seul corpus d'apprentissage. En ce qui concerne le vocabulaire lui-même, qu'il s'agisse de lemmes ou de mots, nous avons entraîné sur ce même corpus des modèles n-grammes et n-lemmes (P#M et P#L), avec $n < 3$. La probabilité de l'étiquette t (Chirac ou Mitterrand) résulte de la combinaison suivante

$$P(t) = \sum_{i=1}^{r} \lambda_i \times p_i(t) \; ; \; \sum_{i=1}^{r} \lambda_i = 1 \tag{4.2}$$

Les valeurs des coefficients λ_i ont été attribuées de façon empirique. L'estimation de ces valeurs a bien entendu été réalisée sur le corpus d'apprentissage. Le poids accordé aux lemmes est deux fois plus important que celui accordé aux mots.

Lorsqu'on utilise des chaînes de Markov en TAL, on est toujours confronté au problème de la couverture des modèles. Le taux de couverture décroît quand augmente l'ordre du modèle. Le problème est bien connu et des solutions de type lissage ou *Back-off* (Manning et Schütze, 1999; Katz, 1987) sont une réponse classique au fait que le corpus d'apprentissage ne suffit pas à garantir une estimation fiable des probabilités. Le problème devenant critique lorsqu'il y a un déséquilibre flagrant entre les deux classes, il nous a semblé inutile, voire contre-productif de calculer des 3-grammes. En nous inspirant des travaux menés en lexicologie sur les discours de Mitterrand, nous avons essayé de prendre en compte certains des traits qualifiés de dominants chez Mitterrand par (Illouz et al., 2000) : adverbe négatif, pronom personnel à la première personne du singulier, point d'interrogation, ou des expressions comme « c'est », « il y a », « on peut », « il faut » (dans les quatre cas, à l'indicatif présent). Ceux-ci ont été traités de la même façon que les autres caractères de la modélisation bayésienne. Après vérification de la validité statistique de ces traits sur le corpus DEFT'05, nous les avons intégrés dans la modélisation mais dans un second temps, nous les avons retirés car même s'ils entraînaient une légère amélioration sur les données de développement, rien ne garantissait qu'il ne s'agissait pas là de tics de langage liés à une période potentiellement différente de celle du corpus de test. Par ailleurs, en cas de portage de

l'application à un autre domaine ou une autre langue, nous ne voulions pas être dépendants d'études lourdes. En tous les cas, nous avons préféré faire confiance aux modèles de Markov pour capturer automatiquement une grande partie de ces tournures.

Modélisation II, sans lemmatisation. Parallèlement et à l'inverse de nos préoccupations précédentes, nous avons souhaité faire fonctionner nos modèles sur le texte à l'état brut, sans enrichissement ou annotation. Pour aller dans ce sens, nous nous sommes demandé à quel point la recherche automatisée des caractéristiques propres à un auteur pourrait être facilitée ou perturbée par le fait de ne pas filtrer ni éliminer quoi que ce soit des discours. Ainsi, nous avons fait l'hypothèse que l'utilisation répétée, voire exagérée de certains termes ne servant qu'à assurer le bâti de la phrase, pouvait prétendre au statut d'indicateur fiable. Pour ce deuxième modèle, nous sommes partis du principe que les techniques de n-grammes appliquées à des tâches de classification, pourraient se passer d'une phase préalable de lemmatisation ou de *stemming*, du rejet des mots-outils et de la ponctuation. Les systèmes n-grammes, (Jalam et Chauchat, 2002; Sahami, 1999) ont montré que leurs performances ne s'améliorent pas après *stemming* ou élimination des mots-outils. Dans cet esprit, nous avons laissé les textes dans leur état originel. Aucun pré-traitement n'a été effectué, même si cette démarche a ses limites : par exemple, « Gasperi » et « Gaspéri » comptent pour des mots différents, qu'il y ait ou non erreur d'accent ; « premier » et « première » sont aussi comptabilisés séparément en absence de lemmatisation. Malgré cela, nous avons voulu donner au modèle un maximum de chances de capturer des particularités de style (manies de ponctuer le texte par l'emploi de telle ou telle personne, de subjonctifs, gérondifs,...) qui sont gommées après application de certains prétraitements comme la lemmatisation. Une classification naïve et un calcul d'entropie ont déjà été rapportés lors de l'atelier DEFT'05 avec un automate légèrement différent (El-Bèze et al., 2005). Seule variante, l'ajout d'une contrainte : tout mot de longueur ≤ 5 n'est pas pris en compte afin d'alléger les calculs. Ceci correspond à un « filtrage » relativement indépendant de la langue.

4.2.2 Automate de Markov

Comme cela était dit en introduction, un discours de Chirac peut avoir fait l'objet de l'insertion d'au plus une séquence de phrases. La séquence M, si elle existe, est d'une longueur supérieure ou égale à deux. Pour prendre en compte cette contrainte particulière, nous avions, initialement, pensé écrire des règles, même si une telle façon de faire s'accorde généralement peu avec les méthodes probabilistes. Dans le cas présent, que faut-il faire si une phrase détachée de la séquence M a été étiquetée M, avec une probabilité plus ou moins élevée (certainement au dessus de 0,5, sinon elle aurait reçu l'étiquette C) ? Renverser la décision, ou la maintenir ? Si l'on opte pour la seconde solution, il serait logique d'extraire également toutes les phrases qui la séparent de la séquence M, bien qu'elles aient été étiquetées C. Mais, dans ce cas, un gain aléatoire en rappel risque de se faire au prix d'une chute de précision.

Pour pouvoir trouver, parmi les chemins allant du début à la fin du discours, celui qui optimise la production globale du discours, nous avons utilisé un automate probabiliste à cinq états (dont un initial I et trois terminaux, C_1, C_2, et M_2). Comme on peut le voir sur la figure 4.1, vers les états dénommés C_1 et C_2 (respectivement M_1 et M_2) n'aboutissent que des transitions étiquetées C (respectivement M). À une transition étiquetée C (respectivement M), est associée la probabilité d'émission combinant pour C (respectivement M) les modèles probabilistes définis en section 4.2.1. Avant de

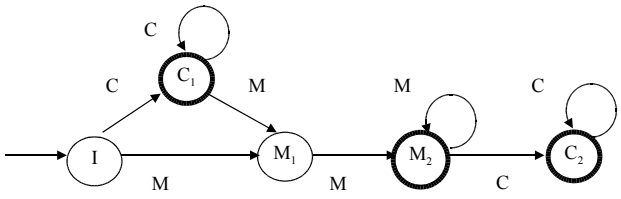


FIG. 4.1: *Machine de Markov exprimant les contraintes générales des trois tâches.*

décrire les étapes ultérieures du processus de catégorisation segmentation, notons que c'est ce composant qui a permis de faire un saut conséquent (plus de 25% en absolu) au niveau des performances et a ouvert ainsi la voie à la mise en place de procédures d'adaptation décrites en section suivante. S'il s'avère qu'étiqueter un bloc de plusieurs segments est plus fiable qu'étiqueter individuellement chaque phrase, il est naturel que cela ait un impact positif sur les performances.

Remarquons par ailleurs que la question aurait pu être gérée autrement, par exemple en utilisant, pour chaque discours, la partie triangulaire supérieure d'une matrice carrée $\Psi[d,d]$ (d étant le nombre de phrases contenues dans le discours en question, voir les figures 4.2 et 4.3). Dans chaque case $\Psi[i,j]$, on calcule la probabilité que la séquence soit étiquetée M entre i et j, et C du début jusqu'au $i-1$ et de $j+1$ à d. Déterminer les bornes optimales de la séquence Mitterrand revient alors à rechercher un maximum sur toutes les valeurs $\Psi[i,j]$ telles que $i > j$. Si cette valeur optimale est inférieure à celle qu'on aurait obtenue en produisant toute la chaîne avec le modèle associé à Chirac, on se doit de supprimer la séquence M. Sauf si on factorise les calculs pour remplir les différentes cases, la complexité de cette seconde méthode est supérieure à celle de l'algorithme de Viterbi (Manning et Schütze, 1999). Il nous a paru néanmoins intéressant d'en faire état dès à présent, car elle offre la possibilité de combiner aisément des contraintes globales plus élaborées que celles que nous prenons en compte dans l'adaptation. Elle peut aussi permettre de mixer des modèles issus de l'apprentissage et d'autres optimisant des variables dédiées à la modélisation de la cohésion interne des séquences qui se trouvent dans le discours traité, et n'ont fait l'objet d'aucun apprentissage préalable, comme nous le montrerons en section 4.3.

4.2.3 Adaptation statique et dynamique

La contrainte de ne pouvoir enrichir le corpus d'apprentissage, sous peine de disqualification, nous a poussé à tirer un parti intégral des données mises à notre dis-

position. Or, en dehors du corpus d'apprentissage, il ne restait plus qu'une issue : intégrer dans l'apprentissage (bien entendu, sans les étiquettes de référence) une partie des données de test. C'est sur ces données que l'adaptation a été pratiquée. Dériver un modèle à partir de l'intégralité des discours de test correspond à ce que nous appelons ici *adaptation statique*. L'*adaptation dynamique*, quant à elle, repose sur un modèle découlant seulement du discours en train d'être testé. Évidemment, il n'est pas interdit de conjuguer les deux approches. Dans un premier temps, nous avions envisagé de pratiquer un étiquetage des données de test, l'objectif étant à l'itération $i+1$ de n'adjoindre au corpus d'apprentissage de X[7] que les phrases s ayant reçu au pas i une probabilité $P_i(X|s)$ supérieure à un certain seuil T_i. Un apprentissage de type maximum de vraisemblance effectué sur les données ainsi collectées peut autant rapprocher qu'éloigner du point optimal. Pour pallier cette difficulté, nous avons opté pour un apprentissage d'*Expectation-Maximisation*, consistant à ne compter pour chaque couple {élément $= e, X$} observé dans les données d'adaptation que la fraction d'unité égale à la probabilité de l'orateur X sachant la phrase qui contient e. La prise de décision repose sur une formule analogue à celle de la formule (4.2). La variable en position 0 est la probabilité de l'étiquette sachant la phrase qui lui a été attribuée à l'itération i. Nous avons fait décroître son poids λ_0 de façon progressive, d'une itération à l'autre par pas de 0,1. Les quatre modèles employés sont, pour les deux premiers, lemmes et mots issus de l'adaptation locale (dynamique), pour les deux derniers, lemmes et mots issus de l'adaptation globale (statique). La pondération entre les différentes probabilités est restée la même durant toutes les itérations : Dynamique {lemmes = 0,4 ; mots = 0,1}, Statique {lemmes = 0,4 ; mots = 0,1}. Les procédures d'adaptation statique et dynamique mises en œuvre durant cette étape ont permis de gagner entre 3 et 4 points de *F-score*.

4.2.4 Réseau de Noms Propres

À partir de la tâche T2, l'ensemble des noms propres était dévoilé aux participants. Établir un lien entre différents éléments apparaissant dans des phrases même éloignées d'un discours donné, nous a paru être un bon moyen pour mettre en évidence une sorte de réseau sémantique permettant aux segments de s'auto-regrouper autour d'un lieu, de personnes et de façon implicite d'une époque. Afin de mixer les relations entretenues entre les noms de pays, leurs habitants, les capitales, le pouvoir exécutif, nous avons complété un réseau fourni par le Centre de Recherche de Xerox[8], en y rajoutant quelques relations issues des Bases de Connaissance que l'équipe TALNE du LIA utilise pour faire fonctionner son système de Questions/Réponses (Bellot et al., 2003).

4.3 Cohésion thématique des discours

En section 4.2.2, nous avions avancé l'hypothèse qu'étiqueter un bloc est plus fiable qu'étiqueter chaque phrase de façon indépendante l'une de l'autre. Cela se discute en

[7] X pouvant prendre ici les valeurs M ou C.
[8] http://www.xrce.xerox.com

fait si on se borne à rechercher la suite de segments qui optimise la cohésion thématique[9] de chacun des deux blocs, il est indispensable de conjuguer cette approche thématique avec un étiquetage d'auteur. Dans cette optique, on peut vouloir trouver un découpage de chaque discours soit en un bloc (*C*...*C*), soit en deux blocs (*C* ...*C*-*M* ...*M*) ou (*M*...*M*-*C*...*C*) soit en trois blocs (*C*...*C*-*M*...*M*-*C*...*C*) tels que le bloc des segments étiquetés *M* ou les blocs (1 ou 2) étiquetés *C* présentent tous les deux une cohérence thématique interne optimale. Pour cela, nous proposons de formaliser le problème comme suit : la probabilité de production d'une phrase est évaluée au moyen d'un modèle appris sur toutes les phrases du bloc auquel elle appartient sauf elle. En maximisant le produit des probabilités d'émission de toutes les phrases du discours, on a toutes les chances de bien identifier des ruptures thématiques. Mais rien ne garantit qu'elles correspondent à des changements d'orateurs. En effet, supposons qu'il n'y ait pas dans un discours donné, d'insertion de phrases de Mitterrand, et que dans les discours de Chirac se trouve une longue digression de 20 phrases. Notre approche risque de reconnaître à tort ces 20 phrases comme attribuables à la classe *M*. Pour éviter ce travers, nous proposons une optimisation mettant en œuvre conjointement les modèles de cohérence interne et ceux issus de l'apprentissage. Nous voyons comment la cohérence interne réussit à renverser presque totalement la situation : ainsi, un gros bloc étiqueté *M* par l'adaptation seule, au sein d'un discours dont la classe est *C* a été étiqueté correctement par la cohérence interne, à l'exception de deux phrases dont les probabilités penchaient trop fortement vers la classe *M*.

Le modèle de cohérence interne cherche donc à maximiser la probabilité d'appartenance des phrases proches des frontières de segments. Il peut utiliser a) le réseau de noms propres et b) la probabilité issue de l'apprentissage par Markov. Pour un discours S_1^d donné, de longueur d, nous cherchons un découpage optimal $\widetilde{D}$ (cf. figure 4.2) et un étiquetage $\widetilde{E}$ tels que

$$(\widetilde{D}, \widetilde{E}) = \text{Arg}\max_{D,E} \left\{ P_I(D, E|S_1^d) \times P'(D, E|S_1^d) \right\} \tag{4.3}$$

où $P'(D, E|S_1^d)$ est la probabilité issue de l'apprentissage et $P_I(D, E|S_1^d)$ la probabilité de cohérence interne (à l'intérieur d'un discours). La conjugaison des modèles d'apprentissage et de cohérence interne est réalisée par le produit entre P' et P_I, qu'il semble légitime de considérer indépendants l'un de l'autre. Comme le découpage ne peut être déduit de l'apprentissage, nous faisons l'hypothèse que $P'(D, E|S_1^d) \cong P'(E|S_1^d)$. Donc

$$(\widetilde{D}, \widetilde{E}) = \text{Arg}\max_{D,E} \left\{ P_I(D, E|S_1^d) \times P'(E|S_1^d) \right\} \tag{4.4}$$

Or, d'après le théorème de Bayes

$$P_I(D, E|S_1^d) = \frac{P(S_1^d|D, E)P(D|E)}{P(S_1^d)} \text{ et } P'(E|S_1^d) = \frac{P'(S_1^d|E)P'(E)}{P'(S_1^d)} \tag{4.5}$$

[9] La cohésion thématique étant un des éléments permettant d'apprécier la cohérence interne d'un discours, nous emploierons de préférence l'expression « cohérence interne » dans la suite.

De ce fait, l'équation (4.4) devient :

$$(\widetilde{D},\widetilde{E}) \cong \operatorname{Arg}\max_{D,E}\left\{\frac{P(S_1^d|D,E)P(D|E)}{P(S_1^d)} \times \frac{P'(S_1^d|E)P'(E)}{P'(S_1^d)}\right\} \tag{4.6}$$

Nous savons que $P(D|E)$ prend toujours des valeurs $\{0, 1\}$ car le découpage est toujours déterminé par les étiquettes (mais pas vice-versa). La probabilité $P'(E)$ ne peut pas être déduite de l'apprentissage (le choix de D peut être considéré comme aléatoire) et $P(S)$ et $P'(S)$ ne dépendent pas de D ou de E. Alors :

$$(\widetilde{D},\widetilde{E}) \cong \operatorname{Arg}\max_{D,E}\left\{P_I(S_1^d|D,E) \times P'(S_1^d|E)\right\} \tag{4.7}$$

Nous avons choisi de représenter un couple (D, E) par un couple de deux indices i

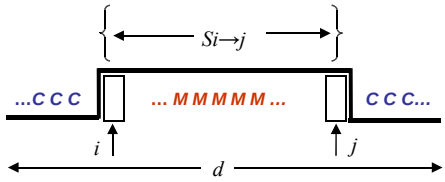


FIG. 4.2: *Schéma de découpage des discours.*

et j dont la signification est donnée par la figure 4.2. Ces deux indices correspondent aux bornes du bloc des segments étiquetés M et à la ligne et la colonne de la matrice Ψ évoquée en section 4.2.2. Pour un discours donné, on aura donc :

$$\Psi[i,j] = P(S_{1\ldots i-1,j+1\ldots d}|C) \times P(S_{i\ldots j}|M) \times P'(S_{1\ldots i-1,j+1\ldots d}|C) \times P'(S_{i\ldots j}|M) \tag{4.8}$$

En faisant l'hypothèse[10] que les segments sont indépendants, nous introduisons le produit sur toutes les phrases du discours en distinguant celles qui sont à l'intérieur du bloc $S_{i\to j}(k = i \ldots j)$ de celles qui sont à l'extérieur $(k = 1 \ldots i-1, j+1 \ldots d)$:

$$\Psi[i,j] = \prod_{k=1\ldots i-1,j+1\ldots d}[P(S_k|\chi) \times P'(S_k|C)] \times \prod_{k=i\ldots j}[P(S_k|\mu) \times P'(S_k|M)] \tag{4.9}$$

où d est la longueur du discours et $\chi = C \setminus S_k$ et $\mu = M \setminus S_k$. Ceci revient à exclure le segment S_k des données qui servent à estimer les paramètres utilisés pour calculer la probabilité de production de ce même segment S_k. Notons que, si les probabilités $P(S_k|\chi) = 1$ et $P(S_k|\mu) = 1$, alors la valeur de $\Psi[i,j]$ est réduite au cas de Markov (adaptation simple). Nous avons exploité la matrice $\Psi[i,j]$ (cf. figure 4.3) en nous limitant à sa partie triangulaire supérieure. Le fait d'exclure la diagonale principale dans les calculs illustre l'exploitation de la contrainte respectée par les fournisseurs du corpus DEFT'05. S'il y a des segments de la classe M insérés, il y a en au moins deux. Le cas des discours étiquetés uniquement C n'est pas représenté dans la figure, mais il a été pris en considération, même s'il n'est pas intégré dans la matrice.

[10] Cette hypothèse va quelque peu à l'encontre de l'objectif recherché, à savoir considérer les segments d'un même bloc comme un tout, mais nous ne savons pas comment faire autrement.

1		...		i	...	d
⋮	•			⋮		
j	•	•		$P(S_i, S_j)$		
	•	•	•	⋮		
⋮	•	•	•	•		
	•	•	•	•	•	⋮
d	•	•	•	•	•	•

FIG. 4.3: *Matrice $\Psi[i, j]$ pour le calcul de la cohérence interne. Les • représentent les cases ignorées pour le calcul des probabilités.*

4.4 Expériences

Pour la Modélisation I, tous les corpus (apprentissage et test) ont été traités par l'ensemble d'outils LIA_TAGG[11]. Dans la phase de développement, le corpus d'apprentissage a été découpé en cinq sous-corpus de telle sorte que pour chacune des cinq partitions, un discours appartient dans son intégralité soit au test soit à l'apprentissage. À tour de rôle, chacun de ces sous-corpus est considéré comme corpus de test tandis que les quatre autres font office de corpus d'apprentissage. Cette répartition a été préférée à un tirage aléatoire des phrases tolérant le morcellement des discours. En effet, un tel tirage au sort présente deux inconvénients majeurs. Le premier provient du fait qu'un tirage aléatoire peut placer dans le corpus de test des segments très proches de segments voisins qui eux ont été placés dans le corpus d'apprentissage. Le second inconvénient (le plus gênant des deux), tient du fait qu'une telle découpe ne permet pas de respecter le schéma d'insertion défini dans les spécificités de DEFT'05.

4.4.1 Résultats de l'adaptation

Une partie des résultats de nos modèles, uniquement avec adaptation, ont été publiés dans les actes du colloque TALN'05. Nous reproduisons ici les observations majeures qui pouvaient être faites sur ces résultats. Le *F-score* s'améliore de façon notable au cours des cinq premières itérations de l'adaptation. Au-delà, il n'y a pas à proprement parler de détérioration mais une stagnation qui peut être vue comme la captation par un maximum local. L'apport des réseaux bâtis autour des noms propres est indéniable (El-Bèze et al., 2005). Nous montrons sur la figure 4.4 les meilleurs *F-score* officiels soumis pour l'ensemble de participants. Le système du LIA senior est positionné, dans les trois tâches, en première place. La méthode de (Rigouste et al., 2005)

[11] Ces outils contiennent : un module de formatage de texte permettant de découper un texte en unités en accord avec un lexique de référence ; un module de segmentation insérant des balises de début et fin de phrase dans un flot de texte (par des d'heuristiques) ; un étiqueteur morphosyntaxique (ECSTA (Spriet et El-Bèze, 1998)) ; un module de traitement des mots inconnus permettant d'attribuer une étiquette morphosyntaxique à une forme inconnue du lexique de l'étiqueteur en fonction du suffixe du mot et de son contexte d'occurrence (basé sur le système DEVIN (Spriet et al., 1996)) et un lemmatiseur.

en deuxième position, utilise quelques méthodes probabilistes semblables aux nôtres, mais ils partent de l'hypothèse que la segmentation thématique est faite au niveau des orateurs (pas au niveau du discours), ils ont besoin de pondérer empiriquement les noms et les dates (tâches T2 et T3), leurs machines de Markov sont plus complexes et il ne font pas d'adaptation, entre autres. Le dévoilement des dates (tâche T3) permet d'améliorer très légèrement les résultats du modèle II, mais entraîne une dégradation sur le modèle I. En ce qui concerne la précision et le rappel au fil des itérations sur l'ensemble de Test (T) ainsi que sur le Développement (D), c'est le gain en précision qui explique l'amélioration due aux Noms Propres. Ce gain allant de pair avec un rappel quasi identique (légèrement inférieur pour le test), il apparaît que le composant Noms Propres fonctionne comme un filtre prévenant quelques mauvaises extractions (mais pas toutes).

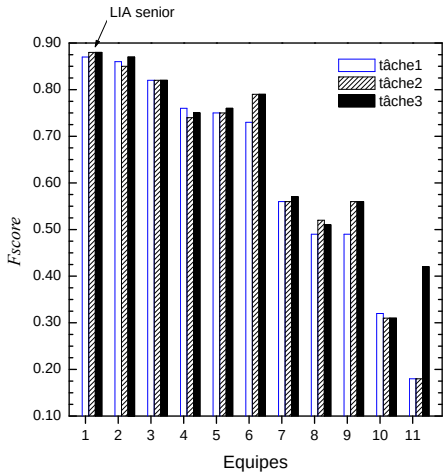


FIG. 4.4: *F-score officiels pour les trois tâches (T1 : pas de noms, pas de dates ; T2 : pas de noms et T3 : avec noms et dates) pour l'ensemble de participants DEFT'05. Les membres des équipes sont cités dans (El-Bèze et al., 2005).*

4.4.2 Résultats avec la cohérence interne

Les résultats ont été améliorés grâce à la recherche d'une cohérence interne des discours. Cette étape intervient après application de l'automate markovien et avant la phase d'adaptation. Nous montrons, sur les figures 4.5 le *F-score* obtenu pour les trois tâches à l'aide d'une adaptation plus la cohérence interne pour les corpus de Développement (D) et de Test (T). Dans tous les cas, l'axe horizontal représente les itérations de l'adaptation. Sur les graphiques, la ligne pointillée correspond aux valeurs du *F-score* obtenues avec l'adaptation seule et les lignes continues à celles de la cohérence interne (une itération : ligne grosse ; deux itérations : ligne fine). Pour les trois tâches, on observe une amélioration notable du modèle de cohérence interne par rapport à celui de l'adaptation seule. Enfin, la valeur la plus élevée de *F-score* est à présent obtenue pour la tâche T3 (0,925). Ce score dépasse largement le meilleur résultat (*F-score* = 0,88)

atteint lors du défi DEFT'05.

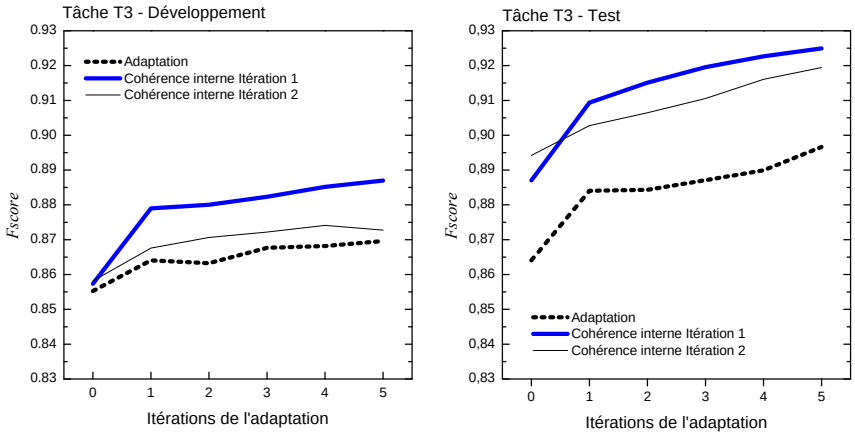


FIG. 4.5: *F-score tâche T3 Modèle I / Adaptation vs Cohérence interne / corpus D et T.*

Les figures 4.6 montrent, avec la même convention que les figures précédentes, le *F-score* pour le modèle II, où n'ont été appliqués ni filtrage ni lemmatisation. Ici encore, les valeurs les plus élevées sont obtenues pour la tâche T3, avec un *F-score* = 0,873. La comparaison avec les performances (*F-score* = 0,801 pour la tâche T3) de ce même modèle que nous avons employé lors du défi DEFT'05, est avantageuse : sept points de plus. Cette amélioration est due essentiellement à la cohérence interne et permet d'approcher la meilleure valeur (*F-score* = 0,881 rapporté dans (El-Bèze et al., 2005)) qui avait été obtenue avec l'adaptation seule et un filtrage et lemmatisation préalables. Bien que l'utilisation de ce modèle soit un peu moins performante (et de ce fait contestée), nous pensons qu'il peut être utile d'y recourir, si l'on veut éviter la lourdeur de certains processus de pré-traitement.

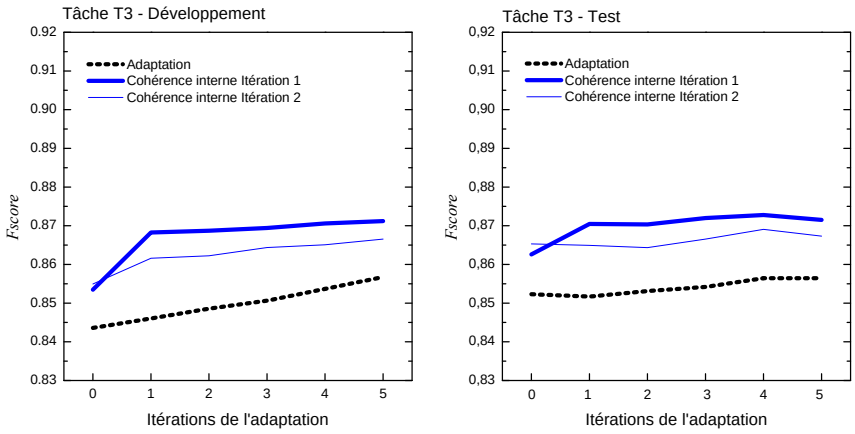


FIG. 4.6: *F-score tâche T3 Modèle II / Adaptation vs Cohérence interne / corpus D et T.*

Le dernier test que nous avons réalisé fait appel à une fusion de l'ensemble des modèles. Nous avons appliqué un algorithme de vote sur presque toutes les hypothèses issues des modèles I et II. Les hypothèses (qui vont faire office de juges) proviennent des différentes itérations de l'adaptation, avec ou sans cohérence interne. Nous avons tenu compte des avis d'un nombre de juges donné (N_J), en pondérant l'avis de chaque juge j par un poids w_j de telle sorte que le critère de décision final soit le suivant

$$\Theta_i = \text{signe}\left(\sum_{j=1}^{N_J} w_j \xi_{i,j} - w_0\right) \tag{4.10}$$

Si Θ_i est négatif alors l'étiquette du segment i sera C; M autrement. Avec $w_j \in \Re$, $\xi_{i,j} \in \{0,1\}$ et la convention 0 = C et 1 = M. La stratégie est la suivante : afin d'avoir un degré de confiance suffisant, il faut retenir les segments auxquels une majorité de juges attribue l'étiquette M. Les paramètres w_j et w_0 ont été ajustés pour minimiser le nombre d'erreurs sur l'ensemble de développement (D). Nous nous sommes proposés de voir cette estimation comme un problème de classification à N_J entrées et une sortie, c'est-à-dire, comme un problème d'apprentissage supervisé. Nous avons ainsi défini un exemple d'apprentissage comme le vecteur binaire $\xi_j = \{0,1\}^{N_J}, j = 1, \cdots, N_J$. La sortie (classe de référence) de cet exemple est un scalaire $\tau = \pm 1$ (-1 classe C, $+1$ classe M). L'ensemble d'apprentissage est donc constitué de S segments et N_J juges, dénoté par $\mathcal{L} = \{\vec{\xi}^{\mu}, \tau^{\mu}\}; \mu = 1, \cdots, S$. Trouver les poids w_j correspond donc à trouver les j poids d'un perceptron entraîné sur l'ensemble $\mathcal{L}$. Nous avons utilisé le perceptron optimal à recuit déterministe[12] presenté dans le chapitre 1, entraîné par l'algorithme Minimerror (Gordon et Berchier, 1993; Torres Moreno et Gordon, 1995), où l'apprentissage garantit que si l'ensemble $\mathcal{L}$ est linéairement séparable, l'algorithme trouve la solution optimale (marge maximale de séparation) et s'il ne l'est pas (comme cela semble être le cas ici), il trouve une solution qui minimise le nombre de fautes commises. Ainsi, nous avons trouvé un seuil w_0 = 8,834 et les poids w_j avec un nombre de juges N_J = 89.

Pour l'ensemble de développement, les résultats obtenus au moyen de cette fusion sont encore meilleurs qu'avec les autres méthodes. Nous obtenons, dans ce cas, un *F-score* = 0,914 avec une précision de 0,916 et un rappel de 0,911. Cependant, pour l'ensemble de test, la fusion ne dépasse pas le meilleur résultat obtenu jusqu'à présent. En effet, on atteint un *F-score* = 0,914, avec une précision de 0,892 et un rappel de 0,937. Il est connu que les perceptrons (et les réseaux de neurones en général) trouvent parfois des valeurs de poids trop bien adaptées à l'ensemble d'apprentissage (phénomène de sur-apprentissage). Le fait de ne pas avoir eu de meilleurs résultats sur l'ensemble de test le confirme. Cependant, nous pensons que si les $\vec{\xi}_i$ étaient des probabilités au lieu d'être des valeurs comprises entre [0,1], on aurait pu observer un meilleur comportement.

[12] La position de l'hyperplan séparateur des classes se fait par une modification progressive des poids (descente en gradient) contrôlés au moyen d'une température de recuit lors de l'apprentissage.

4.4.3 Analyse des erreurs

Nous avons analysé les erreurs commises par notre système. Sur un total de 27 163 phrases de l'ensemble de Test de la tâche T3, le Modèle I avec la méthode d'adaptation et la cohérence interne des discours, a fait un total de 578 erreurs (*F-score* = 0,925) :

- 233 erreurs de la classe C (faux négatifs assimilés au rappel), dont : 37 phrases C à la frontière inversée ($\approx$ 16%) ; 113 phrases C en blocs ($\approx$ 49%) ; 83 phrases C entre blocs C ($\approx$ 36%) ;
- 345 erreurs de la classe M (faux positifs assimilés à la précision), dont : 35 phrases M à la frontière inversée ($\approx$ 10%) ; 126 phrases M en blocs ($\approx$ 37%) ; 184 phrases M insérées dans 21 discours de classe C exclusive ($\approx$ 53%).

Le problème le plus grave concerne la précision (59% du total des erreurs), et ici, la plus grande majorité (53% de faux positifs) est due aux insertions des phrases M dans des discours de classe C[13]. L'autre problème se présente dans les 126 phrases en blocs inversés (37%). Ces problèmes sont peut-être dus à l'utilisation de la cohérence interne : en adaptation seule , la précision est toujours plus élevée que le rappel (en D comme en T). Pour la cohérence interne, la situation est inversée : le rappel est bien meilleur que la précision. Le même comportement a été retrouvé dans le Modèle II. Un autre pourcentage important d'erreurs (49% de faux positifs) a lieu dans l'inversion d'un nombre important de blocs (113 phrases). Enfin, une autre partie non négligeable (10% de faux positifs, 16% de faux négatifs) correspond à l'inversion de catégorie d'une phrase unique à la frontière des découpages (soit i ou j, voir figure 4.2). La détection de cette frontière, reste un sujet très délicat avec nos approches.

4.5 Conclusions et perspectives

La formalisation de la cohérence interne des discours (El-Bèze et al., 2007) a beaucoup amélioré nos résultats precédents (El-Bèze et al., 2005). Cette cohérence ainsi que l'adaptation ont été combinées conjointement avec les modèles d'apprentissage préalablement développés, comme la modélisation bayésienne qui semble déterminant, l'automate de Markov et des processus d'adaptation. Les résultats obtenus pour la tâche T3 avec la cohérence interne en terme de *F-score* = 0,925 sont très encourageants. Cependant, l'utilisation de cette cohérence présente un risque : quelques phrases avec une thématique différente, peuvent faire basculer tout un bloc vers l'autre étiquette. Ce type de comportement local entraîne des instabilités globales dont la prévision reste très difficile, ayant comme conséquence une baisse générale des performances. Ne pas lemmatiser et ne rien filtrer dégrade un peu les performances mais permet d'éviter l'application d'un processus additionnel de pré-traitement qui pour certaines langues est relativement lourd. La fusion des hypothèses vue comme un vote de plusieurs juges pondérés par un perceptron optimal a permis de surpasser les résultats précédents en développement (*F-score* = 0,914). Cependant nous pensons qu'il reste encore du travail pour améliorer cette stratégie afin d'obtenir de meilleures performances en test.

[13] Voir (El-Bèze et al., 2007) pour l'analyse du discours 520, concernant cette situation problématique.

Des études comme celle de (Rigouste et al., 2005) sur le même corpus confirment que l'utilisation de méthodes probabilistes est la mieux adaptée à ce type de segmentation thématique. Le recours à un réseau de Noms Propres est utile et nous encourage par la suite à employer une ressource lexicale comme (Vossen, 1998) pour tirer parti de réseaux sur les noms communs. Pour s'affranchir des contraintes liées à la constitution d'une ressource sémantique, il serait judicieux de recourir à des approches telles que LSA (Deerwester et al., 1990) ou PLSA (Hofmann, 1999). D'autres perspectives d'application, comme celle de la séparation de thèmes sont aussi envisageables. Il faut reconnaître, cependant, que s'il s'il avait été question de traiter un texte moins artificiel que celui proposé par DEFT, par exemple un dialogue, la difficulté aurait été accrue. Des frontières thématiques ne coïncident pas forcément avec des débuts de phrase. Les thèmes peuvent s'entremêler et composer un tissu discursif où les fils sont enchevêtrés de façon subtile. Beaucoup reste à faire pour pouvoir différencier plusieurs orateurs ou plusieurs thèmes comme envisagé dans le cadre du Projet Carmel (Chen et al., 2005).

Chapitre 5

Cortex *es Otro Resumidor de TEXtos*

S'il est un homme tourmenté par la maudite ambition
de mettre tout un livre dans une page,
toute une page dans une phrase,
et tout une phrase dans un mot,
c'est moi.
Joseph Joubert (1754-1824). PENSÉES, ESSAIS.

Lors du rigoureux hiver canadien de 2001, sans la moindre intention d'aller à l'extérieur à une température de -40°C et surchargé de cours à préparer dans l'Université du Québec à Chicoutimi, je me demandais s'il n'y avait pas un moyen simple de synthétiser les textes de mes cours pour épargner du temps. Face à moi j'avais toujours une pile de livres et d'articles de systèmes digitaux et systèmes d'exploitation qui ne m'inspiraient pas beaucoup la lecture. Si la machine serait capable de les lire à ma place puis de me présenter une synthèse plus courte, donc plus *humaine*... (*tout un livre dans une page*) Mais comment faire... ? Je l'ignorais, mais je commençais à me poser le problème du résumé automatique de textes. J'arrivais donc dans ce domaine d'une façon hasardeuse et trop naïve. Je n'étais pas au courant de l'état de l'art sur les condensés de textes, mais j'ai commencé a définir, avec Patricia Velázquez, les briques de base d'un système très naïf de résumé par extraction de phrases. J'avais déjà une certaine expérience avec le modèle vectoriel de textes et j'ai décidé donc de l'appliquer aux résumés automatiques. De ces journées sombres et enneigées du nord du Canada est sorti l'idée de COrtex, un autre Résumeur de TEXtes. Je montrerai en plus de Cortex, basé sur une approche numérique, une méthode hybride linguistique-numérique pour générer des resumés d'un domaine spécialisé.

Mes travaux sur Cortex ont commencé à l'UQAC, puis à l'École Polytechnique de Montréal et continués au LIA d'Avignon. Les idées originales ont étés publiées dans les conférences ARCo'01 (Torres-Moreno et al., 2001) et JADT'02 (Torres-Moreno et al., 2002). Les travaux sur un resumeur hybride utilisant Cortex et des méthodes linguistiques ont été publiés dans le congrès MICAI'07, (da Cunha Iria et al., 2007).

5.1 Introduction

L'information textuelle sous forme électronique s'accumule rapidement et en très grande quantité. Les documents sont catégorisés d'une façon très sommaire. Le manque de standards est un facteur critique. Ainsi, l'analyse automatique de texte (le dépistage, l'exploration, l'extraction d'information, le résumé, etc.) sont des taches extrêmement difficiles (Manning et Schütze, 1999). Les méthodes linguistiques sont pertinentes dans ces tâches, mais leur utilisation concrète demeure encore difficile (en raison de plusieurs facteurs tels que l'ampleur ou la dynamique des corpus) ou limitée à des domaines restreints.

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 (ANSI, 1979). Produire un résumé pertinent demande au résumeur (humain ou système) l'effort de sélectionner, d'évaluer, d'organiser et d'assembler des segments d'information selon leur pertinence. Cette pertinence (Mani et Mayburi, 1999; Morris et al., 1999; Mani, 2001) peut être guidé par un sujet ou une thèmatique en particulier, comme on verra lors du prochain chapitre.

Dans son étude sur les résumeurs humains professionels (Cremmins, 1996) relève une *analyse locale* (contenu dans une phrase) et une autre *Globale* (contenu à travers les phrases). Cremmins recommande entre 8-12 min pour résumer un article scientifique type : ce qui est beaucoup moins du temps nécessaire pour véritablement le comprendre ! De nos jours le résumé automatique est un sujet de recherche très important. Introduit par Luhn (Luhn, 1958) à la fin des années 50, avec un système de résumé par extraction de phrases, le résumé automatique est un processus qui transforme un texte source en texte cible, de taille plus réduite et dans lequel l'information pertinente est conservée. Des techniques qui utilisent la position textuel (Edmundson, 1969; Brandow et al., 1995; Lin et Hovy, 1997), les modèles Bayesienes (Kupiec et al., 1995), la pertinence marginale maximale (Goldstein et al., 1999) ou la structure de discours ont été utilisées. La plupart des travaux sur le résumé par extraction des phrases appliquent les techniques statistiques (analyse de fréquence, recouvrement de mots, etc.) aux unités telles que les termes, les phrases, etc. D'autres approches sont basées sur la structure du document (mots repère, indicateurs structuraux) (Edmundson, 1969; Paice, 1990a), la combinaison de l'extraction de l'information et de la génération de texte, l'utilisation des SVM (Mani et Bloedorn, 1998; Kupiec et al., 1995) pour trouver des patrons dans le texte, les chaînes lexicales (Barzilay et Elhadad, 1997; Stairmand, 1996) ou encore la théorie de la structure rhétorique (Mann et Thompson, 1987).

Mes recherches ont porté sur l'obtention des résumés informatifs par extraction de phrases pertinentes. Je présenterai Cortex, un systeme basé sur la représentation vectorielle des textes (Salton et McGill, 1983; Salton, 1989), sous forme d'une chaîne de traitement numérique qui combine plusieurs traitements statistiques et informationnels (tels que les calculs d'entropie, le poids fréquentiel des segments et des mots, et plusieurs mesures de Hamming parmi d'autres) avec un algorithme optimal de décision. Cortex génère des condensés de textes par extraction des phrases pertinentes.

5.2 Cortex

Le système Cortex est composée de deux algorithmes : une méthode de construction des métriques informationnelles indépendantes, couplée avec un algorithme qui combine l'information venant des métriques. Ce dernier prends une décision sur la pertinence des segments en fonction d'une stratégie de vote. Un pré-traitement de textes avec des processus classiques est applique aux documents avant de les traiter. Sur la figure 5.1, je montre l'architecture modulaire de Cortex. Je détaillerai par la suite les algorithmes qui le composent.

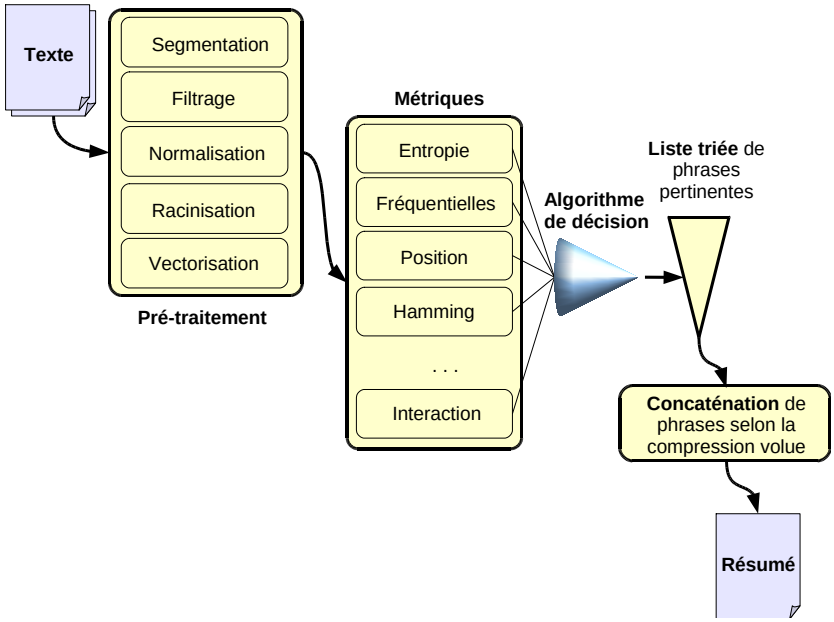


FIG. 5.1: *Architecture générale de Cortex.*

Dans l'approche vectorielle on traite de textes dans leur ensemble, en passant par une représentation numérique très différente d'une analyse structurale linguistique mais qui permet des traitements performants (Manning et Schütze, 1999). L'idée consiste à représenter les textes dans un espace approprié et à leur appliquer des traitements vectoriels. Cortex comporte un ensemble des processus de filtrage, segmentation et lemmatisation. Par opposition à l'analyse symbolique classique, ces processus sont très performants et peuvent être appliqués sur des gros corpus. Le texte original comporte N_M mots (mots fonctionnels, noms, verbes fléchis...). J'emploie la notion de *terme* pour designer un mot plus abstrait (Manning et Schütze, 1999). Pour réduire la complexité des processus de réduction du lexique sont amorcés[1]. La lemmatisation de verbes dans les langues à morphologie variable (langues romanes) s'avère très important pour la réduction du lexique, et consiste à trouver la racine des verbes fléchis et à ramener les

[1]La suppression des mots fonctionnels, des mots à haute et très basse fréquence d'apparition, suivi par la suppression (facultative) du texte entre parenthèses, de chiffres et des symboles.

mots au singulier masculin avant de les compter[2]. Ce processus permet de diminuer la malédiction dimensionnelle qui pose de sérieux problèmes dans des grandes dimensions[3]. La segmentation par phrases est naïve, et realisée en utilisant les marqueurs forts (tels que <!>,<?>,<.>,<:>)[4]. Un indice de repérage important d'information est le titre d'un document. Toutefois la plupart de nous expériences ont été réalisées sur de textes bruts, donc les titres, sous-titres et sections ne sont pas marqués explicitement. Après le pré-traitement, le nouveau texte comporte P segments avec N_f termes totaux.

La vectorisation transforme un document dans un ensemble de P vecteurs $\vec{\sigma} = (s_1, s_2, \cdots, s_{N_M})$. Chaque segment du texte est représenté par un vecteur $\vec{\sigma}$. La dimension N_M est le nombre total de mots type. Seulement les termes à fréquence supérieure à 1 seront utilisés (Torres-Moreno et al., 2001, 2002), et donc un lexique de $N_{\mathcal{L}}$ termes gardé pour obtenir une matrice Terme-Segment $S = \vec{\sigma}_\mu; \mu = 1, \cdots, P$, qui représente le lexique réduit du texte :

$$S = \begin{pmatrix} s_1^1 & s_1^2 & \cdots & s_1^{N_{\mathcal{L}}} \\ s_2^1 & s_2^2 & \cdots & s_2^{N_{\mathcal{L}}} \\ \vdots & \vdots & \ddots & \vdots \\ s_P^1 & s_P^2 & \cdots & s_P^{N_{\mathcal{L}}} \end{pmatrix} ; \quad s_\mu^i = \begin{cases} TF^i & \text{si le terme } i \text{ existe} \\ 0 & \text{autrement} \end{cases} \tag{5.1}$$

où chaque composante $\vec{\sigma} = (s_1, s_2, \cdots, s_{N_{\mathcal{L}}})$ contient la fréquence s_μ^i du terme i dans un segment μ. Cette matrice contient l'information fréquentielle essentielle du texte.

La matrice binaire ξ est derivée de S :

$$\xi_\mu^i = \begin{cases} 1 & \text{si le terme } i \text{ existe} \\ 0 & \text{autrement} \end{cases} \tag{5.2}$$

Dans cette matrice chaque composante montre la présence ($\xi_i^\mu = 1$) ou l'absence ($\xi_i^\mu = 0$) du mot i dans un segment μ. Les algorithmes de Cortex vont agir sur ces deux matrices, qui constituent l'espace des entrées du système. Les segments possèdent évidement une quantité hétérogène d'information : la tâche consiste alors en repérer les segments les plus importants pour obtenir un extract du document.

[2] Ainsi on pourra ramener à la même forme **chanter** les mots *chantaient, chanté, chanteront* et eventuellement *chante* et *chanteur*.

[3] J'ai exploré aussi d'autres voies pour la réduction du lexique en utilisant des processus de synonimisation à l'aide de dictionnaires, mais il y a un compromis sensible concernant l'introduction de bruit, qui n'est pas facile à trouver. Les résultats ne sont donc pas concluants.

[4] Le critère de segments à taille fixée a été écarté, car on cherchait l'extraction des phrases complètes.

5.3 Algorithmes

5.3.1 Métriques

Des informations mathématiques et statistiques importantes sont calculées à partir des matrices S (5.1) et ξ (5.2) sous forme de métriques. Elles mesurent la quantité d'information contenue dans chaque segment : plus le segment est important, plus les valeurs qui lui sont associées sont élevées.

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

$$F^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} s_{\mu}^{i} \tag{5.3}$$

    s_i^{μ} est la fréquence du mot i dans le segment μ. L'expression :

$$T = \sum_{\mu=1}^{P} \sum_{i=1}^{N_{\mathcal{L}}} s_{\mu}^{i} \tag{5.4}$$

    correspond à la taille du lexique fréquentiel contenue dans σ. Nous introduisons ici la quantité :

$$\rho_{\sigma} = \frac{T}{N_M} \tag{5.5}$$

    définie comme le ratio de réduction du lexique fréquentiel.
  - Interaction de segments. Dans chaque segment μ, un mot i qui est présent au même temps dans un ou plusieurs autres segments, on dit qui est en « interaction ». La somme de toutes les interactions de mots de chaque segment constitue alors l'interaction entre segments. Nous la comptabilisons de la façon suivante :

$$I^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \sum_{\substack{j=1 \\ j \neq \mu}}^{P} \xi_{i}^{j} \tag{5.6}$$

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

$$p_i = \frac{1}{T} \sum_{\mu=1}^{P} s_{\mu}^{i} \tag{5.7}$$

    La somme fréquentielle des probabilités est calculée :

$$\Delta = \sum_{i=1}^{N_{\mathcal{L}}} p_i s_{\mu}^{i} \; ; \text{si } \xi_{\mu}^{i} \neq 0 \tag{5.8}$$

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

$$E^{\mu} = -\sum_{i=1}^{N_{\mathcal{L}}} x_{\mu}^{i} \log_2 x_{\mu}^{i} \tag{5.9}$$

avec

$$x_i^{\mu} = \frac{s_i^{\mu}}{\sum_{i=1}^{N_{\mathcal{L}}} s_{\mu}^{i}} \tag{5.10}$$

– Mesures de Hamming. Une distance de Minskowski a été utilisée comme mesure de base.
  – La matrice de Hamming H. Une matrice carrée de $N_L \times N_L$ où chaque case $H[i,j]$ représente le nombre de phrases utilisant exclusivement un des termes i ou j.

$$H_{m,n} = \sum_{j=1}^{N_P} \left\{ \begin{array}{ll} 1 & \text{si } \xi_{j,m} \neq \xi_{j,n} \\ 0 & \text{autrement} \end{array} \right\} \quad \begin{array}{l} m \in [2, N_L] \\ n \in [1, m] \end{array} \tag{5.11}$$

  La matrice de Hamming est une matrice triangulaire inférieure où l'index m représente la ligne et l'index n la colonne, correspondant à l'index des mots ($m > n$). L'idée principale est que si deux mots importants (qui peuvent être des synonymes) sont dans la même phrase, cette phrase doit certainement être importante. L'importance de chaque paire de mots correspond directement à la valeur dans la matrice de Hamming H.
  Les distances de Hamming Ψ. Cette quantité mesure la distance entre les paires de mots i et j dans l'espace des segments. Chaque mot étant représenté par un vecteur binaire $\vec{\xi}_i = \{0,1\}^P$. Il faut d'abord, calculer la matrice de Hamming H, qui est une matrice diagonale supérieure à dimension N.

$$H_i^{i+1} = \left\{ \begin{array}{l} 1\,;\, \text{si } \xi_{\mu}^{i} \neq \xi_{\mu}^{j} \\ 0 \text{ autrement} \end{array} \right\} \begin{array}{c} i=1,\cdots,N_{\mathcal{L}}-1 \\ j=i+1,\cdots,N_{\mathcal{L}} \\ \mu=1,\cdots,P \end{array} \tag{5.12}$$

  Ensuite on calcule la somme des distances de Hamming :

$$\Psi^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \sum_{j=i+1}^{N_{\mathcal{L}}} H_i^j \text{ si } \left(\xi_i^{\mu}, \xi_j^{\mu}\right) \neq 0 \tag{5.13}$$

  – Le poids de Hamming des segments. Chaque segment possède un « poids » ϕ^{μ}, qui est égal à la somme des termes présents dans le segment, c'est-à-dire, dans chaque ligne de la matrice ξ (5.1) :

$$\phi^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \xi_{\mu}^{i}\,;\, \text{si } \xi_{\mu}^{i} \neq 0 \tag{5.14}$$

  – La somme des poids de Hamming de mots Θ. De la même manière, on peut mesurer le poids spécifique des mots sur chaque colonne μ de la matrice $\xi; \mu = 1, \cdots, P$, ce qui donne un « Poids de Hamming des mots » :

$$\psi^{i} = \sum_{\mu=1}^{P} \xi_{\mu}^{i}\,;\, \text{si } \xi_{\mu}^{i} \neq 0\,;\, i = 1, \cdots, N_{\mathcal{L}} \tag{5.15}$$

Ensuite on calcule la somme des poids de Hamming des mots par segments :

$$\Theta^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \psi_i \, ; \text{ si } \xi_{\mu}^{i} \neq 0 \tag{5.16}$$

– Mesures mixtes. Des mesures de distances combinées avec des mesures frequentielles ont été aussi considérées.
  – Le poids de Hamming lourd. Il est obtenu de la multiplication du poids de Hamming du segment ϕ par le poids de Hamming des mots S_{HM} :

$$\Pi^{\mu} = \phi^{\mu} S_{HM}^{\mu} \tag{5.17}$$

  – Somme des poids de Hamming de mots par fréquence Ω. Ceci correspond à la somme des poids de Hamming des mots i existants dans chaque segment μ, multipliée par la fréquence correspondante.

$$\Omega^{\mu} = \sum_{i=1}^{N_{\mathcal{L}}} \phi^{\mu} s_{\mu}^{i} \, ; \text{ si } s_{i}^{\mu} \neq 0 \tag{5.18}$$

– Structure simple. La métrique X est basée sur la structure empirique de certains documents (comme les textes journalistiques) : les phrases les plus importantes se trouvent au début et à la fin du document. Nous avons modélisé cette métrique comme une parabole :

$$X^{\mu} = (\mu - P\%2)^2 \tag{5.19}$$

La matrice de Hamming pourrait être utilisée pour éviter la redondance car elle mesure la utilisation de termes exclusifs dans deux phrases.

5.3.2 Algorithme de décision

Nous avons développé un algorithme pour récupérer l'information codée par les métriques. L'idée est simple : 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). Une première version a été publiée en (Torres-Moreno et al., 2001)[5]. L'AD a par la suite été modifié par une autre version plus adaptée.

[5]Il utilise deux probabilités mutuellement exclusives : p_0 et p_1. On présente les k votants en modifiant p_0 et p_1 en fonction des sorties π_j; $j = 1, \cdots, k$ (les valeurs des métriques normalisées préalablement).
– Pour chaque segment $\vec{\xi}^{\mu}$; $\mu = 1, 2, \cdots, P$
– $p_0^{(0)} \leftarrow p_1^{(0)} \leftarrow \frac{1}{2}$
– Pour $j = 1, \cdots, k$; votants ν
  – Le votant ν_j calcule une valeur π_j normalisée entre (0,1) qui reflète l'importance du segment μ.
  – Si la valeur π_j est significative ($\neq \frac{1}{2}$) on modifie les probabilités p_0 et p_1 à l'itération $t+1$ en

L'algorithme de décision combine les sorties normalisées de toutes les métriques (comprises entre [0,1]) dans une forme sophistiquée afin de calculer le score de chaque phrase s. Deux moyennes sont calculées : la tendance positive, où $\lambda_s > 0,5$, et la tendance négative, où $\lambda_s < 0,5$ (le cas $\lambda_s = 0,5$ est ignoré[6]). Pour calculer cette moyenne, on divise par le nombre total de métriques Γ et pas uniquement par le nombre d'eléments positifs ou négatifs (la moyenne réelle des tendances). En divisant par Γ, on a devéloppé un algorithme plus décisif que la simple moyenne et plus realiste que la moyenne réelle des tendances. Voici l'algorithme de décision qui combine le vote des métriques :

$$\sum^{s} \alpha = \sum_{v=1}^{\Gamma} (||\lambda_s^v|| - 0.5); \;\; ||\lambda_s^v|| > 0,5 \tag{5.22}$$

$$\sum^{s} \beta = \sum_{v=1}^{\Gamma} (0.5 - ||\lambda_s^v||); \;\; ||\lambda_s^v|| < 0,5 \tag{5.23}$$

v est l'indice de la metrique, $\sum_{s}^{\Gamma}$ est la somme des différences absolues entre $\|\lambda\|$ et 0,5, $\sum^{s} \alpha$ sont les métriques *positives* normalisées, $\sum^{s} \beta$ les métriques *négatives* normalisées et Γ le nombre de métriques utilisées. La valeur attribuée à chaque phrase est calculée au moyen de : Λ^s est la valeur utilisée pour la décision de retenir ou non la phrase s. À

if $\underline{(\sum^s \alpha > \sum^s \beta)}$ **then**
$\Lambda^s = 0,5 + \sum^{s} \alpha/\Gamma$: retenir la phrase s
else
$\Lambda^s = 0,5 - \sum^{s} \beta/\Gamma$: éliminer la phrase s
end

la fin, N_P phrases sont triées selon leur valeur $\Lambda^s; s = 1, \cdots, N_P$.

fonction des valeurs à l'itération t :

$$\text{Si } \left(\pi_j \leq \frac{1}{2}\right) \quad \text{alors} \quad p_0^{(t+1)} \leftarrow p_0^{(t)}(1-\pi_j); p_1^{(t+1)} \leftarrow p_1^{(t)}\pi_j \tag{5.20}$$

$$\text{sinon} \quad p_1^{(t+1)} \leftarrow p_1^{(t)}(1-\pi_j); p_0^{(t+1)} \leftarrow p_0^{(t)}\pi_j \tag{5.21}$$

– À la fin de la présentation des k votants, on mesure les probabilités de décision :

Si $(p_0 > p_1)$ alors AD = $2|p_0 - p_1|$ et il retient le segment μ
sinon AD = 0 et il élimine le segment

Il possède deux propriétés intéressantes : convergence et amplification. Les probabilités p_0 et p_1 sont modifiées en tout temps de façon mutuellement exclusive, et leur écart est changé toujours avec une probabilité $\geq 0,5$ de l'améliorer. Il amplifie car la probabilité de choisir un segment pertinent est $\geq \pi_j$ (meilleur votant branché à ce moment).

[6]La moyenne simple peut être ambigüe si la valeur est proche de 0,5, ainsi l'algorithme de décision élimine les phrases dont leur score = 0,5.

Avant que l'algorithme de décision ne soit utilisé, chacune des métriques de Cortex μ doit être normalisée afin d'éviter les différentes plages de valeurs. La metrique resultante est notée par $\hat{\mu}$ et définie par :

$$\hat{\mu}(d) = \frac{\mu(d) - m}{M - m} \tag{5.24}$$

où : $m = \min\{\mu(d) : d \in \Delta(T)\}$; $M = \max\{\mu(d) : d \in \Delta(T)\}$.

5.4 Évaluation des résumés

L'évaluation des résumés reste une tâche très difficile et subjective. Malgré les efforts de la communauté TAL, on n'a pas réussi à la faire automatiquement. Elle peut être réalisée en évaluant indépendamment le résumé (façon intrinsèque) ou en évaluant le résumé de forme indirecte dans une tâche précise, comme par exemple, couplé avec un système de Questions-Réponses (façon extrinsèque) (Mani et Mayburi, 1999). Les résumés sont aussi évalués soit manuellement soit semi-automatiquement. La première méthode consomme un coût de temps humain élévé (chaque résumé doit être lu, évalué et validé) et reste très subjective, car la divergence entre les juges peut être considérable. La deuxième méthode est plus standard et a la capacité d'être reproductible, mais elle exige un nombre de résumés de référence produits par des humains.

Différentes approches existent pour l'évaluation semi-automatique des résumés, telles que ROUGE (Lin, 2004), *Pyramids* (Passonneau et al., 2005) ou *Basic Elements* (BE) (Hovy et al., 2005). Les mesures ROUGE se sont imposées actuellement[7] comme évaluation des résumés.

*Recall-Oriented Understudy for Gisting Evaluation (*ROUGE*).* Elle mesure le rappel de n-grammes des mots entre un résumé candidat et un ensemble de résumés référence (Lin, 2004). ROUGE-2 est basée sur les bigrammes des mots, est définie par l'équation 5.25. $Count_{match}$ représente le nombre maximum des bigrammes en co-occurence entre un résumé candidat et un ensemble de résumés référence R_S. Dans ROUGE-2 le dénominateur de l'équation est la somme total du nombre de bigrammes presents dans les résumés référence.

$$\text{ROUGE-2} = \frac{\sum_{s \in R_S} \sum_{bigram \in s} Count_{match}}{\sum_{s \in R_S} \sum_{bigram \in s} Count} \tag{5.25}$$

SU4 est également un rappel de bigrammes de mots mais étendue, ce qui permet de considérer les bigrammes et les trous arbitraires dans une longueur maximale de 4. Par exemple, la phrase « *pourquoi utilise-t-on le résumé* » possède $Count(4,2) = 6$ bigrammes à trou : « *pourquoi utilise-t-on* », « *pourquoi le* »,

[7] Les trois approches ont été retenues lors de *Document Understanding Conferences* (DUC), qui concernent les résumés guidés par une thématique (voir chapitre 6).

« pourquoi résumé », *« utilise-t-on le »*, *« utilise-t-on résumé »* et *« le résumé »*. Nous avons calculé la somme de bigrammes avec un trou arbitraire γ par

$$Count(k,n) = C\binom{n}{k} - \sum_{0}^{k-\gamma}(k-\gamma)\,;\,\gamma \geq 1 \tag{5.26}$$

où n est la longueur du n-gramme et k la longueur de la phrase en mots.

Basic Elements (BE). Est une méthode d'évaluation spécifique qui utilise les unités plus petites du contenu, appelées *Basic Elements*, pour pallier certaines défaillances des n-grammes (Hovy et al., 2006). Le problème de l'évaluation ROUGE est que des unités multi-termes (telles que « Les États Unis Mexicains ») ne sont pas traitées en tant qu'unités, biaisant ainsi le score et que des mots relativement sans importance (tel que « Les ») ont le même poids que d'autres mots relativement plus importants. L'évaluation *Basic Elements* tente de résoudre ces problèmes en utilisant un analyseur syntaxique pour extraire uniquement les unités sémantiques minimales valides.

Ces mesures, combinées avec la méthode *Pyramids* et la lecture directe des résumés seront beaucoup utilisées dans le chapitre suivant pour évaluer les systèmes de résumé multi-document guidés par une thématique.

5.5 Expériences et discussion

Avant l'année 2002 les évaluations ROUGE, *Basic Elements* ou *Pyramids* n'existaient pas. Á l'époque j'ai donc procedé à une évaluation indirecte, en comparant les résumés générés par nos systèmes avec les extraits produits par un certain nombre de juges humains. Une sorte de mesure de rappel et précision. Je reproduis dans la figure 5.2 cette évaluation émpirique, qui a été rapportée dans (Torres-Moreno et al., 2001, 2002) concernant des textes en français et en espagnol. J'ai décidé d'inclure des mesures ROUGE, afin d'être moins empirique dans l'évaluation de Cortex. Des textes en français et en anglais ont été chosis, tel que rapporté dans (Fernandez et al., 2007a). Recémment, des tests en langue somalienne ont été realisés. Pour cela un stemming élémentaire a été utilisé. Le lecteur intéressé peut s'adresser à (Abdillahi et al., 2006).

5.5.1 Évaluations empiriques

Nous avons testé notre algorithme sur des articles de vulgarisation scientifique. Les textes extraits de la presse sur Internet sont de petite taille. L'objectif a été d'obtenir un extrait représentant 25% du nombre de segments total. Nous avons comparé les performances de Cortex avec celles du système Minds[8], du logiciel Copernic Summarizer[9] et du synthétiseur Word de Microsoft. Nous avons aussi demandé à 14 personnes de

[8] http://messene.nmsu.edu/minds/SummarizerDemoMain.html
[9] http://www.copernic.com

constituer un condensé à la main, c'est-à-dire : de choisir les phrases du texte qui leur semblaient les plus pertinentes[10].

– **Textes en français.** Nous avions déjà étudié le texte « Puces » [11], mélange hétéroclite de sujets puces biologiques et informatiques, donc *artificiellement* ambigu, dans le cadre de la classification de texte par leur contenu (Torres-Moreno et al., 2000). Il contient 29 phrases et 653 mots. Les segments les plus importants sélectionnés par les humains sont le 2, 5, 15 et 17 (figure 5.2b). Une partie de nos résultats est montrée sur la figure (5.2a), où on voit que les segments importants ont été bien repérés. Cortex et Minds obtienent un résumé équilibré (même si ce dernier ne trouve ni le segment 5 ni le 15). Par contre les résultats de Word sont biaisés et peu pertinents comme on le voit sur la même figure. Pour le texte « Fêtes » [12] les

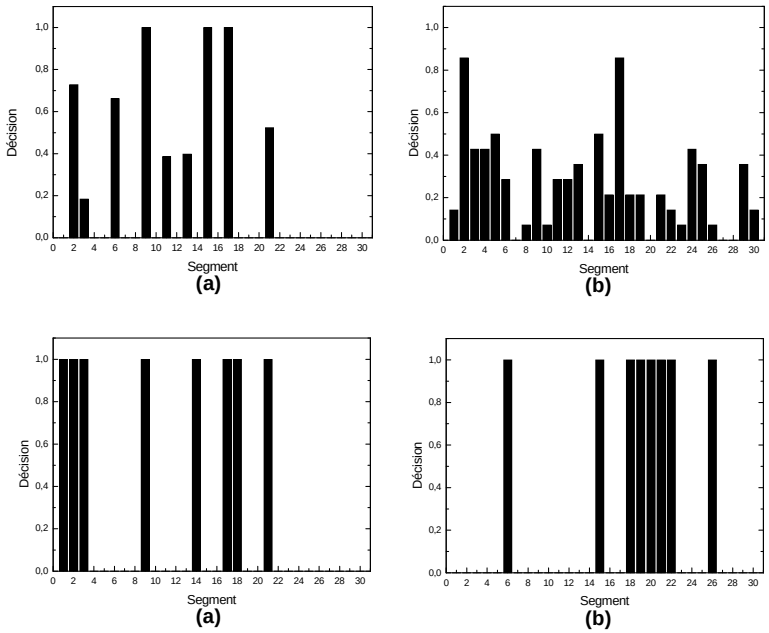


FIG. 5.2: *Choix de segments pertinents pour le texte « Puces » par en haut a) Cortex : les segments 2 et 17 ont été bien choisis et b) Par les 14 sujets humains. En bas, a) Minds et b) synthétiseur Word.*

résultats montrent que Cortex trouve des résumés acceptables. Nous avons effectué des comparaisons avec Copernic Summarizer (Huot, 2000), et nos condensés sont comparables voire de meilleure qualité. Dans d'autres tests les condensés trouvés par Cortex semblent être assez cohérents (Torres-Moreno et al., 2001).

– **Textes en espagnol.** Nous avons étudié deux textes techniques en espagnol : « Nopal » et « Tabaco » [13]. On constate la bonne qualité du condensé, même dans des

[10] Tous les sujets avaient un niveau d'études universitaire, habitués à rédiger des résumés.
[11] http://www.lia.univ-avignon.fr/chercheurs/torres/recherche/cortex
[12] http://www.quebecmicro.com/6-12/6-12-28.html
[13] Textes de l'année 2000, recupérables à l'adresse http://www.invdes.siw.com.mx

textes avec un lexique redondant.

Nous avons toujours constaté que les condensés des humains dépendent de l'expertise des personnes et des leurs capacités d'abstraction. Ceci produit souvent des résultats très différents entre les juges. Mais malgré cela, les choix des humains nous à toujours semblé être une référence. Les mesures ROUGE l'ont confirmé plus tard.

5.5.2 Évaluations avec Rouge

- **Textes en français.** Pour ces tests, nous avons choisi les textes : « 3-mélanges »[14] composé de trois thématiques (27 phrases, 826 mots, 8 références), « puces » (29 phrases, 653 mots, 8 références) et « J'accuse »[15], lettre d'Émile Zola (206 phrases, 4 936 mots, 6 références).
- **Textes en anglais.** Trois textes de l'encyclopédie collaborative Wikipédia en anglais ont été analysés : « Lewinsky »[16] (30 phrases, 816 mots, 7 références), « Québec »[17] (44 phrases, 1 190 mots, 8 références) et « Nazca »[18] (52 phrases, 1 310 mots, 6 références).

Nous avons evalué les résumés en utilisant les mesures rappel de ROUGE-2 et SU4 (c.f. Section 5.4). Le ratio de compression est variable suivant le nombre de phrases des textes. Résumés au 25% : « 3-mélanges », « puces », « Quebec » et « Nazca », résumé au 12% : « J'accuse » et résumé au 20% : « Lewinsky ». Le tableau 5.1 montre les performances moyennes du rappel ROUGE de Cortex vs. les systèmes MEAD, Copernic Summarizer, Pertinence[19], Microsoft Word et une *baseline* très simple où les phrases ont été choisies au hasard. La version en ligne du système MEAD[20] produit uniquement des résumés en anglais, d'où les symboles ⊘ au tableau. En gras, sont affichées

Corpus	**MEAD**		**Copernic**		**Word**		**Cortex**		*Baseline*	
	R2	SU4	R2	SU4	R2	SU4	R2	SU4	R2	SU4
3-mélanges	⊘	⊘	0,4231	0,4348	*0,4301*	*0,4376*	**0,4967**	**0,5064**	0,3074	0,3294
Puces	⊘	⊘	**0,5775**	**0.5896**	0,1656	0,1955	*0,5356*	*0,5588*	0,3053	0,3272
J'accuse	⊘	⊘	0,2235	0,2707	*0,3140*	*0,3441*	**0,6316**	**0,6599**	0,2177	0,2615
Lewinsky	0,4756	0,4744	*0,5580*	*0,5610*	0,1412	0,1581	**0,6183**	**0,6271**	0,2767	0,2925
Quebec	*0,4820*	*0,5169*	0,4492	0,4859	0,3615	0,4028	**0,5636**	**0,5872**	0,2999	0,3524
Nazca	*0,4446*	*0,4671*	0,4270	0,4495	0,3498	0,3588	**0,5894**	**0,5966**	0,2999	0,3524

TAB. 5.1: *Rappel* ROUGE-2 *(R2) et SU4 des résumés génériques. Résumés au 25% : 3-mélanges, puces, Québec et Nazca ; résumé au 20% : Lewinsky et résumé au 12% : J'accuse.*

les performances les plus élevées et en italique celles en deuxième position (tous scores

[14] Recupérable à l'adresse `http://www.lia.univ-avignon.fr`
[15] Téléchargeable à l'adresse `http://www.cahiers-naturalistes.com/jaccuse.htm`
[16] `http://en.wikipedia.org/wiki/Monica_Lewinsky`
[17] `http://en.wikipedia.org/wiki/Quebec_sovereignty_movement`
[18] `http://en.wikipedia.org/wiki/Nazca_Lines`
[19] `http://www.pertinence.net`
[20] Accessible sur `http://tangra.si.umich.edu/clair/md/demo.cgi`

confondus). On constate que Cortex est un système de résumé automatique très performant (il obtient 10 premières places et 2 deuxièmes). Cela est vrai pour tous les textes sauf pour « puces », où Copernic Summarizer est legèrement supérieur. Mais même dans ce cas, les performances de Cortex ne sont pas du tout mauvaises. Microsoft Word est souvent proche, voire pire que la *baseline*, ce qui d'ailleurs ne devrait pas étonner. Le graphique 5.3 présente les moyennes des moyennes pour chaque méthode. Par soucis de clarté, je montre uniquement l'écart type correspondant à SU4 (axe vertical).

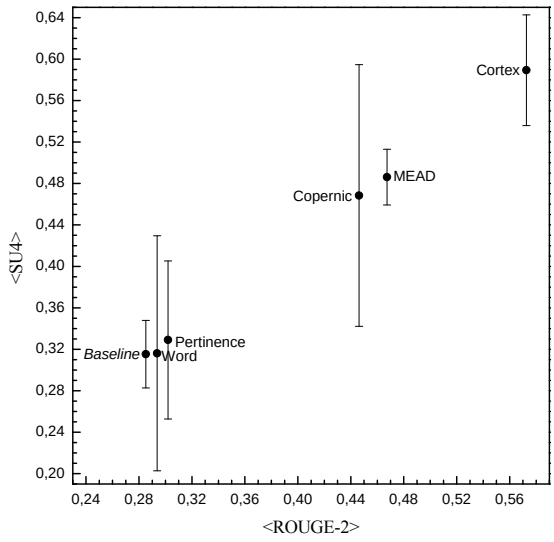


FIG. 5.3: *Moyenne des scores moyens de rappel pour tous les textes.*

Au chapitre 8 je présenterai un comparatif sur ces mêmes textes, contre une méthode de résumé automatique basée sur une nouvelle mesure de similarité de documents, l'*énergie textuelle*.

5.5.3 Discussion

Taille du lexique. Nous avons défini :

$$\rho_{\mathcal{L}} = \frac{N_{\mathcal{L}}}{N_M} \tag{5.27}$$

comme le ratio de réduction du lexique filtré/lemmatisé.

Nous savions que grâce au processus de pré-traitement, le lexique était de plus en plus réduit, c'est-à-dire $N_{\mathcal{L}} \leq N_f \leq N_M$. Des études sur les ratios de réduction moyens du lexique ont été effectués sur des corpus en français et en espagnol. Ceci nous a permis d'établir expérimentalement des estimateurs $\hat{\rho}_{\mathcal{L}}$ et $\hat{\rho}_s$ pour le lexique filtré/lemmatisé réduit $\rho_{\mathcal{L}}$ (5.27) et le lexique fréquentiel ρ_s (5.5) respectivement. Si nous introduisons :

$$\hat{N}_M = \frac{1}{\tau}\sum_{i=1}^{\tau} N_{Mi}; \hat{N}_f = \frac{1}{\tau}\sum_{i=1}^{\tau} N_{fi}; \hat{N}_{\mathcal{L}} = \frac{1}{\tau}\sum_{i=1}^{\tau} N_{\mathcal{L}i}; \hat{T} = \frac{1}{\tau}\sum_{i=1}^{\tau} T_i;$$

Alors :

$$\hat{\rho}_f = \frac{\hat{N}_f}{\hat{N}_M} \; ; \; \hat{\rho}_s = \frac{\hat{T}}{\hat{N}_M} \; ; \; \hat{\rho}_{\mathcal{L}} = \frac{\hat{N}_{\mathcal{L}}}{\hat{N}_M}$$

Nous avons calculé : $\hat{\rho}_f = 0,414 \pm 0,032$; $\hat{\rho}_{\mathcal{L}} = 0,068 \pm 0,015$ et $\hat{\rho}_s = 0,224 \pm 0,058$. La réduction de la taille du lexique filtré/lemmatisé $\hat{\rho}_{\mathcal{L}}$ suit un comportement linéaire (Torres-Moreno et al., 2002) par rapport au nombre de mots du texte original, donc pour obtenir un condensé d'un texte avec nos méthodes on utilise seulement un seizième du volume de termes totaux du document.

Pouvoir discriminatoire des métriques. Il est mesuré en fonction de leurs écart-types par rapport au choix des segments. En effet, il y a des métriques que sont plus discriminantes que d'autres. La somme des distances de Hamming entre mots à l'intérieur d'un segment est très discriminante. Par contre, les métriques qui impliquent des calculs d'entropie ou de valeurs fréquentielles semblent l'être moins. La figure 5.4 montre les moyennes du pouvoir discriminatoire des métriques.

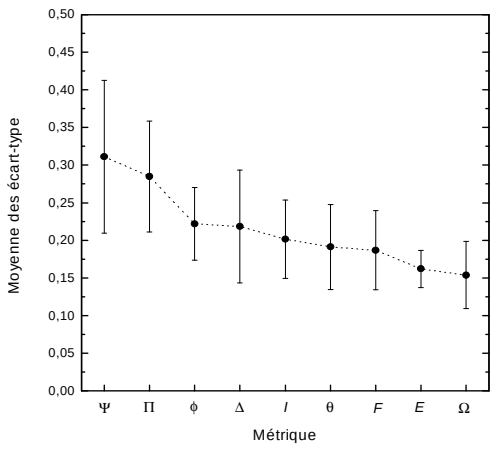


FIG. 5.4: *Moyenne du pouvoir de discrimination des métriques.* **1.** *Les distances de Hamming* Ψ, **2.** *Les poids de Hamming lourd* Π, **3.** *Le poids de Hamming des segments* ϕ, **4.** *La somme des probabilités par fréquence* Δ, **5.** *Les interactions* I, **6.** *La somme des poids de Hamming* Θ, **7.** *La fréquence* F, **8.** *L'entropie* E *et* **9.** *La somme fréquentielle des poids de Hamming* Ω.

Cependant la combinaison des métriques est un facteur plus determinant que l'ordre des métriques. Pour le moment, en résumé générique, nous avons decidé d'utiliser toutes les métriques disponibles. Cette étude de l'impact des métriques sur le résumé générique sera completée par deux autres : au moment de guider le résumé par une thématique (c.f. chapitre 6) et par une autre concernant le raffinement de requêtes (c.f. chapitre 7).

Ordre de présentation des segments. Nos expériences ont montré que l'ordre de présentation des segments n'a aucune influence sur le choix final de l'Algorithme de décision. Nous avons découpé les textes en segments et nous les avons mélangé au hasard pour obtenir un nouveau texte. Ce texte a été présenté à nouveau à

Cortex (sans la métrique X) et la même pondération des phrases a été retrouvée. Mais bien évidemment, le résume produit par ce mélange est incohérent, ce qui est assez logique.

5.6 Et si la linguistique pouvait... ? une approche hybride

Jusqu'à maintenant mes algorithmes utilisaient peu ou pas de ressources linguistiques. Le système Cortex, bien que performant, plafonne car il est confronté au niveau de granularité de la phrase et il est incapable de traiter les dépendances fines telles que les anaphores. Une discussion entre Iria Da Cuhna, Eric SanJuan et moi a portée sur la possible combinaison de méthodes numériques telles que Cortex et les approches linguistiques développés au IULA[21]. Nos expériences se sont focalisées sur le résumé de textes médicaux. Ce sujet est un domaine très important avec une très grande quantité d'information à traiter (Afantenos et al., 2005). L'objectif de ce travail est d'aider à résoudre ce problème. Nous nous sommes intéressées à analyser les techniques employées pour le résumé automatique des textes dans les domaines spécialisés, spécifiquement des domaines scientifiques-techniques. À l'avenir nous pensons prolonger ce travail à d'autres domaines tels que la chimie, la biochimie, la physique, la biologie, la génomique, etc. Nous travaillons avec des articles médicaux parce que ce genre de texte est publié dans des journaux avec leurs résumés correspondants écrits par les auteurs. Nous les utilisons pour évaluer les résumés de nos systèmes. Une autre motivation pour mener à bien ces travaux est que, bien qu'il y ait beaucoup de systèmes de résumé automatique utilisant la statistique (Barzilay et Elhadad, 1997; Kupiec et al., 1995; Silber et McCoy, 2000) ou la linguistique (Alonso et Fuentes, 2003; Marcu, 1998; Ono et al., 1994; Teufel et Moens, 2002), il y en a très peu combinant les deux critères (Alonso et Fuentes, 2003; Aretoulaki, 1996, 1994; Nomoto et Nitta, 1994). Notre idée est que, pour générer un bon résumé, on a besoin d'utiliser les aspects linguistiques des textes, mais aussi de profiter des avantages des techniques statistiques. Sur la base de cette idée, nous avons développé un système hybride qui bénéfice des différents aspects des textes afin de générer des résumés. Nous avons intégré trois modèles dans ce système : Cortex et Enertex (basés sur le modèle vectoriel, couplés à l'extracteur de termes Yate) et Disicosum (modèle linguistique).

Le système de résumé linguistique utilisé fait partie de la thèse d'Iria Da Cunha. Le système Yate a été devéloppé par Jorge Vivaldi. Tous les deux chercheurs à l'IULA. Cette étude est le fruit d'une collaboration en 2007 entre le IULA et le LIA.

Cortex. Il a été utilisé avec toutes les métriques, sans réglage particulier des paramètres.

Enertex. (Fernandez et al., 2007a) approche inspirée par la physique statistique adapté aux problèmes du TAL. L'algorithme modèle les documents comme un réseau de

[21] Institut de Linguistique Appliquée `http://www.iula.upf.edu`, Université Pompeu Fabra (Barcelone).

neurones dont l'énergie textuelle est étudiée. L'idée principale est qu'un document peut être traité comme un ensemble d'unités (les mots) qui interagissent les unes avec les autres. Je détaillerai cette méthode au chapitre 8.

Yate. Les termes extraits représentent des « concepts » appartenant au domaine médical et leur *termicité* modifiera leurs poids dans la matrice terme-segment de Cortex. Yate (Vivaldi, 2001; Vivaldi et Rodríguez, 2002) est un outil d'extraction de termes candidats dont les caractéristiques principales sont l'utilisation d'une combinaison de plusieurs techniques d'extraction de termes et l'utilisation de EWN[22], une ontologie lexico-sémantique.

Disicosum. L'hypothèse de base est que les professionnels des domaines spécialisés (spécifiquement, le domaine médical) utilisent des techniques concrètes pour résumer leurs textes (da Cunha et Wanner, 2005). (da Cunha et Wanner, 2005, 2006) ont étudié un corpus contenant des articles médicaux et leurs résumés afin de trouver quelle information devrait être choisie pour obtenir un résumé spécialisé. Un autre point de départ est l'idée que différents types de critères linguistiques devraient être utilisés pour avoir une bonne représentation des textes, et exploiter ainsi leurs avantages. Les systèmes de résumé automatique basés sur la linguistique utilisent, en générale, un seul type de critère (les termes (Luhn, 1958) ; la position textuelle (Brandow et al., 1995; Lin et Hovy, 1997) ; la structure discursive (Marcu, 1998; Teufel et Moens, 2002), etc.), mais pas leurs combinaisons. Disicosum est constitué de règles à l'égard des structures textuelles, lexicales, discursives, syntaxiques et communicatives :

- **Règles textuelles :** i) Le résumé doit contenir l'information de chaque section de l'article[23]. ii) Les phrases de certaines positions doivent être bonifiées d'un poids supplémentaire[24].
- **Règles lexicales :** a) Augmenter le score des phrases contenant : i/ Mots du titre principal (sauf les mots fonctionnels), ii/ Formes verbales à la 1ère personne du pluriel, iii/ Mots d'une liste contenant les verbes (*analyser*, *observer*, etc.) et les substantifs (*but*, *objectif*, *résumé*, *conclusion*, etc.) qui pourraient être pertinents, iv/ Toute information numérique des sections Patients et méthodes et Résultats. b) Éliminer les phrases contenant : i/ Références à des tables/figures (les modèles linguistiques montrent que seulement une partie de la phrase pourrait être éliminée : *comme montré dans le Tableau...*, *sur la figure 4 nous pouvons observer...*), ii/ Références à des aspects statistiques/informatiques : *informatique*, *programme*, *algorithme*, *coefficient*, etc., iii/ Références à des travaux précédents : *et al.* et quelques modèles linguistiques, par exemple, « déterminant + nom (travail | étude | recherche | auteur) ». Exceptions : *cette étude*, *notre recherche...* iv/ Références à des définitions : *c'est/Ils sont défini par/comme...*.

[22]EWN (www.illc.uva.nl/EuroWordNet) est une prolongation multilingue de WordNet (wordnet.princeton.edu), une ontologie lexico-sémantique. L'unité sémantique de base dans les deux ressources est le « synset » qui groupera ensemble plusieurs mots simples/multi termes qui peuvent être considérés des synonymes dans quelques contextes. Les *synsets* sont liés à l'aide d'étiquettes sémantiques. En raison de la polysémie, une simple entrée lexicale peut être attachée à plusieurs « synsets ».

[23]Les articles considerés ont les sections : Introduction, Patients et méthodes, Résultats et Conclusions.

[24]Les 3 premières phrases de la section d'Introduction, les 2 premières phrases de Patients et méthodes et de Résultats, et les 3 premières et les 3 dernières de la section Conclusions.

– **Règles discursives et règles combinant les structures discursives, syntaxiques et communicatives**. Deux cadres théoriques ont été utilisés : la théorie de la structure rhétorique (RST) (Mann et Thompson, 1988) et la théorie de la signification du texte (MTT) (Mel'cuk, 1988, 2001). RST est une théorie de l'organisation du texte qui caractérise sa structure comme un arbre hiérarchique contenant les relations discursives (Elaboration, Concession, Condition, Contraste, Union, Evidence, etc.) entre ses éléments, appelés noyau et satellite. MTT est une théorie qui intègre plusieurs aspects de la langue. Dans ce travail, d'une part, nous avons utilisé sa conception de la syntaxe profonde des dépendances, qui représente une phrase comme un arbre où les unités lexicaux sont les nœuds et les relations entre elles sont marquées comme Actants et relations Attributive, Appenditive et coordonnatrices. D'autre part, nous avons utilisé la distinction entre Thème et Rhème, qui font partie de la structure communicative de MTT. Quelques exemples de ces règles sont[25] :
  – SI S est satellite$_{CONDITION}$ C alors maintenir S
    [Si ces patients ont besoin d'un flux supérieur,] S [c'est probablement qu'il n'est pas bien toléré.] N
  – SI S est satellite$_{BACKGROUND}$ B alors éliminer S
    ~~[Les personnes qui ne veulent pas manger et avec un complexe de grosses ont l'anorexie.]~~ S [Nous avons étudié l'aspect des complications dans les patients anorexiques.] N
  – SI S est satellite$_{ELABORATION}$ E1 et S élaborent sur le thème du noyau d'E1 alors éliminer S
    [Les personnes qui ne veulent pas manger et avec un complexe de grosses ont l'anorexie.] N ~~[Un des problèmes de ces patients est le manque d'amour propre.]~~ S
  – SI S est satellite$_{ELABORATION}$ E1 et S sont ATTR alors éliminer S
    [Ils ont selectioné 274 contrôles,] N ~~[qui hypothétiquement auront eu les mêmes facteurs de risque.]~~ S

Pour l'implantation des règles textuelles et lexicales nous avons utilisé un segmenteur pour l'espagnol[26] et TreeTagger (Schmid, 1994). Cependant nous avons eu des problèmes pour l'exécution complète du modèle. D'abord, aucun analyseur syntaxique est capable d'obtenir la structure discursive de textes en espagnol[27]. Ensuite, il n'existe aucun analyseur syntaxique pour obtenir la structure communicative. Il y a seulement quelques publications à ce sujet, comme par exemple (Hajicova et al., 1995). Enfin, bien qu'il y ait quelques analyseurs syntaxiques de dépendances pour l'espagnol (Attardi, 2006; Asterias et al., 2005), leurs résultats ne sont pas fiables. Ainsi une solution intermédiaire consiste à simuler la sortie de ces analyseurs. Un taggeur XML syntaxique-communicatif et discursif semi-automatique a été conçu afin d'étiqueter les textes. Une interface d'annotation, où l'utilisateur choisi la relation entre les différents éléments (noyau et satellites) des textes, a été développée. Il est fait en deux étapes. D'abord, l'utilisateur détecte les relations entre les phrases, puis il trouve des relations à l'intérieur des phrases (si elles existent). Le résultat est une représentation du texte sous la forme d'un arbre de relations.

[25] N=noyau, S=satellite. On éliminera le texte souligné.
[26] Développé à l'IULA.
[27] Il y a pour l'anglais (Marcu, 1998, 2000) et un projet en cours pour le portugais (Pardo et al., 2004).

Une approche hybride numérique-linguistique

La figure 5.6 présente l'architecture du système hybride. D'abord le système applique les règles d'élimination (c.f. section 5.6) sur le texte original afin d'obtenir une réduction de ≈ 20-30% de sa longueur. Sur ce texte réduit, on applique séparément les systèmes Cortex, Enertex et Disicosum. Un algorithme de décision traite les sorties normalisées des systèmes comme suit : soit il choisi les phrases selectionées par les trois systèmes, soit il retient les phrases choisies par deux systèmes ou finalement, faute de consensus, l'algorithme accorde la priorité à celles dont les scores sont les plus grands. En cas d'égalité de score, on garde les phrases selon leur ordre d'apparition.

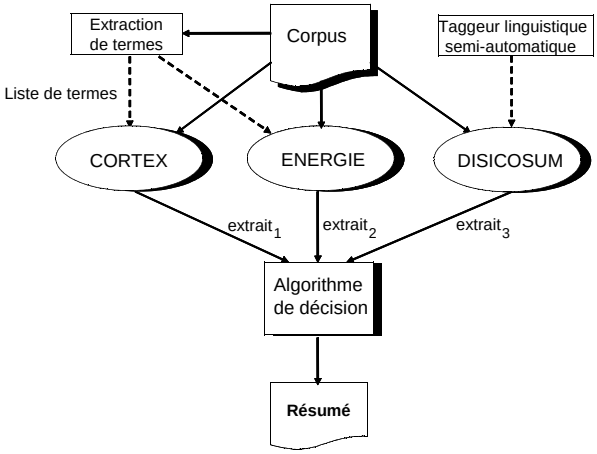


FIG. 5.5: *Architecture du système hybride de résumé.*

Une petite expérience et discussion

Le corpus utilisé contient des articles médicaux en espagnol du journal *Medicina Clínica*[28]. Les textes ont été préalablement semi-automatiquement étiquetés (avec l'interface) par 5 personnes. Ensuite, nous avons comparé les résumés produits par les trois systèmes, en leur donnant comme entrée les articles réduits par les règles d'élimination (c.f. 5.6). En outre nous avons crée deux baselines afin de les inclure dans l'évaluation. Les deux font une selection aléatoire des phrases de chaque section considerée séparément. La différence est que la Baseline_1 a été faite à partir de l'article original et la Baseline_2 à partir de l'article réduit par l'application des règles d'élimination (5.6). Pour évaluer les résumés, nous les avons comparés aux résumés ecrits par les auteurs, en utilisant ROUGE (Lin, 2004). Pour l'application de ROUGE, nous avons utilisé une lemmatisation en espagnol et une *stoplist*. Pour calculer la longueur des résumés, nous avons obtenu le nombre de phrases moyen de chaque section présentes dans les résumés de l'auteur. Puis, nous avons décidé d'inclure dans nos résumés une phrase

[28] http://db.doyma.es/cgi-bin/wdbcgi.exe/doyma/mrevista_info.sobre?pident_revista=2

aditionnelle par section. Cette décision a été prise parce que nous nous sommes aperçu que généralement les auteurs fusionnent les phrases du texte dans les résumés.

L'évaluation des résultats est montrée au tableau 5.6. Nous avons utilisé les mesures

	ROUGE-2		SU4	
	Mediane 1	**Mediane 2**	**Mediane 1**	**Mediane 2**
Système Hybride	**0.3638**	**0.3808**	**0.3613**	**0.3732**
Disicosum	**0.3572**	**0.3956**	**0.3359**	**0.3423**
Cortex	**0.3324**	0.3324	**0.3307**	**0.3255**
Enertex	0.3105	0.3258	0.3155	0.3225
Cortex texte intègral	0.3218	**0.3329**	0.3169	0.3241
Enertex texte intègral	**0.3598**	**0.3598**	**0.3457**	**0.3302**
Baseline_1	0.2539	0.2688	0.2489	0.2489
Baseline_2	0.2813	0.3246	0.2718	0.3034

TAB. 5.2: *Comparaison des valeurs* ROUGE *entre different résumés.*

ROUGE malgré le fait que seulement un résumé de référence a été fourni. Néanmoins, ces mesures fournissent une forme standard de comparaison et garantissent la reproductibilité de nos expériences. Pour évaluer l'impact de la qualité du taggeur semi-automatique sur la performance de Disicosum, 20% des documents ont été étiquetés dans un temps restreint (30 minutes par texte) et les autres sans restrictions de temps. Par conséquent on s'attend à ce que la concordance du taggeur linguistique sur les 80% des textes soit meilleure que pour les 20% étiquetés dans un temps restreint.

Le tableau montre les medianes de chaque système sur le corpus (Médiane 1) et sur l'ensemble des documents reduits étiquetés sans restrictions de temps (Médiane 2). Les polices de caractères dépendent du quartile (grandes polices pour les grands quartiles, plus petits pour les autres). D'après la mediane des documents pour lequels l'étiquetage a été fait dans un temps sans restriction, les résumés de Disicosum semblent être les plus proches du résumé de l'auteur selon ROUGE-2. Selon la mesure SU-4, Disicosum est le meilleur parmi les systèmes individuels, mais le système hybride est le plus performant de tous. En ce qui concerne l'ensemble de tous les textes, les medianes de Disicosum sont inférieurs aux précédents. Cependant ils restent parmi les plus hauts par rapport aux systèmes individuels. Ceci montre que la qualité de l'étiquetage du modèle linguistique a un impact direct sur la qualité du résumé même si l'étiquetage a été effectué indépendamment du but du résumé. Les systèmes Cortex et Enertex ont été testés directement sur les textes complets et sur les textes segmentés par sections indépendantes. Le deuxième meilleur système individuel selon ces résultats semble être Enertex sur le texte intégral. Il s'avère qu'Enertex fonctionne mieux sans l'indication de considérer que le résumé devrait contenir éléments venant de chaque section du texte. Une explication pourrait être qu'Enertex compare toutes les phrases deux à deux. Plus le texte contient de phrases, mieux sera la représentation vectorielle. Cortex utilise des critères locaux puisqu'il a été construit pour résumer efficacement les corpus volumineux. Sur des textes courts, le manque de mots fréquents réduit la performance du système. Cependant il s'avère ici qu'il peut prendre en compte les propriétés structu-

rales des textes. En ce qui concerne le système hybride, les expériences montrent qu'il améliore la proximité par rapport au résumé de l'auteur dans tous les cas à l'exception de ROUGE-2 si l'on considère l'étiquetage linguistique humain sans restriction de temps.

Finalement, nous avons effectué une autre expérience : un même texte ettiqueté par 5 personnes différentes et les résumés produits par Disicosum utilisés comme modèles de référence ont été utilisés pour calculer les mesures ROUGE des autres systèmes. L'idée était de trouver quel système est le plus proche au modèle linguistique. Cortex et Enertex se sont revelés comme les plus proches de ce modèle. En d'autres termes, les performances de Disicosum et celles des deux résumeurs numériques sont équivalentes. En outre, nous montrons comme l'utilisation d'un système d'extraction de termes peut coopérer avec les méthodes numériques pour la tâche de résumés. D'une part, nous avons montré que la combinaison de méthodes statistiques et linguistiques afin de développer un système hybride pour le résumé automatique donne des bons résultats, encore meilleurs que ceux obtenus par chaque méthode séparément. Finalement, nous avons trouvé que le système Disicosum obtient des résumés très semblables, bien que différents annotateurs étiquettent le texte (c'est-à-dire, les annotateurs produisent différents arbres de discours). D'autres tests, comparant plusieurs résumés produits par des médecins et le système hybride, doivent être réalisés. Des applications dans d'autres domaines et langues sont également considérées.

5.7 Bilan et perspectives

J'ai toujours eu le sentiment qu'un titre pertinent, correspond au résumé *maximal* d'un texte. Il doit être court, porteur d'information et concis. Trouver un bon titre pour un texte est déjà quelque chose de complexe. La tâche de résumé automatique de textes a fait un chemin. Il est loin d'être fini. Les systèmes se heurtent au difficile problème de la compréhension du texte. Tant qu'on sera incapable de le résoudre, le résumé automatique sera une simple approximation du résumé humain. Mais le résumé humain, lui, n'est pas facile à caractériser non plus. Une étude menée en collaboration avec le Laboratoire des Sciences de l'Éducation à Grenoble (LSE)[29] sur un nombre important de sujets ($\approx$ 215) divisé en groupes de niveau éducatif différent (4éme, 3éme, 2éme, 1er, CAP et Master 2) a montré la non concordance entre les personnes[30]. D'où la difficulté de l'évaluation. Cependant, les méthodes numériques (Cortex et Enertex) ont montré une bonne corrélation avec les sujets les plus aisés dans la production de résumés (c'est à dire les étudiants universitaires du Master). Une publication conjointe entre le LIA et le LSE est en cours de soumission à JADT'08. En plus, de nombreuses expériences montrent que les résumeurs professionnels n'ont même pas besoin de comprendre le texte. D'apres cette hypothèse, une machine peut toujours tenter de s'approcher de cette tâche. C'est justement cela que j'ai essayé de faire avec Cortex : pousser au maximum la démarche vers le tout numérique. Cortex est un résumeur de textes très performant.

[29] http://web.upmf-grenoble.fr/sciedu

[30] Ceci avait déjà été constaté dans plusieurs travaux, entre autres par (Yousfi-Monod et Prince, 2006).

Cet algorithme permet de traiter de vastes corpus, relativement indépendamment de la langue, sans préparation, avec une certaine quantité de bruit, de manière dynamique et en un temps court. Plusieurs tests faits en comparaison avec des sujets humains ou d'autres méthodes de résumé automatique, ont montré que Cortex retrouve les segments de texte les plus pertinents (indépendamment de la taille du texte et des thématiques abordées). On obtient ainsi un résumé balancé car la plupart des thèmes sont abordés dans le condensé final. Copernic Summarizer communique avec l'utilisateur en lui demandant des concepts à retenir dans le résume. Ceci est une approche intéressante qui sera explorée au chapitre suivant. L'algorithme de décision basé sur le vote de métriques est robuste, convergent, amplificateur et indépendant de l'ordre de présentation des phrases. Nous pensons que l'ajout d'autres métriques (comme l'entropie résiduelle, la détection des changements d'entropie, maximum d'entropie) pourraient améliorer la qualité des condensés. En particulier, une nouvelle métrique de similarité, dérivée de l'énergie textuelle qui sera présentée au chapitre 8, s'avère déjà très intéressante. Un identificateur automatique de langues, à base d'uni-grammes de lettres a été incorporé au système. Il permet la detéction de l'anglais, l'espagnol, le français, l'allemand (et même le somali).

Maintenant parlons du rôle des poids des termes, ce qui a servi au modèle hybride linguistique-numérique. Les termes peuvent être pondérés par des mécanismes classiques de Tf.Idf, ou d'autres plus complexes, comme ceux d'un extracteur de termes. Un test exploratoire a été réalisé dans le cadre de résumés en espagnol dans le domaine spécialisé (médecine), en utilisant un extracteur de termes comme Yate. Il semblerait que cela aide à mieux repérer les phrases pertinentes, mais des tests supplémentaires devraient le confirmer (ou infirmer). La fusion de méthodes numériques avec une approche linguistique a montré que cette voie est très intéressante car elle produit des résumés plus proches de ceux attendus par un utilisateur.

La réponse alors a la question *Et si la linguistique pouvait ... ?* est oui. La linguistique ajoute une valeur de finesse aux méthodes numériques, et, on obtient comme sous-produit évident, des performances améliorées. Je montrerai l'utilisation d'autres modules linguistiques en post-traitement au chapitre suivant et une combinaison symbolique-numérique pour le raffinement de requêtes au chapitre 7. La production de résumés génériques est une tâche très importante en TAL, mais qui peut être plus intéressante si les résumés sont personnalisés par les besoins de l'utilisateur. Je me suis intéressé à ce type de résumés, qui sont guidés par une thématique qui peut être précise ou floue. L'adaptation de Cortex à ces tâches fera l'objet des deux chapitres suivants.

Chapitre 6

Résumé guidé par une thématique

Rien ne résume un homme,
pas même ses idées.
Mourad Bourboune. Le Mont des genêts

Ce chapitre présente une approche pour le résumé automatique multi-documents guidé par une thématique (ou résumé personnalisé). On a étudié l'efficacité de combiner un système de résumé générique avec l'information venant d'un corpus en entier et celle des documents prise individuellement. Je présenterai Neo-Cortex, un système de résumé multi-documents basé sur le système Cortex introduit précédemment. Des expériences sur les données de *Document Understanding Conferences* (DUC) 2005, 2006 et 2007 ont prouvé que Neo-Cortex est un système efficace, obtenant des bonnes performances sur la tâche principale de résumé multi-documents, guidé par une thématique. La combinaison de plusieurs systèmes de résumé automatique du LIA (développés par Benoît Favre, Laurent Guillard, Patrice Bellot, Frédéric Béchet, Marc El-Bèze, Florian Boudin et moi-même) par un système de fusion, a montré des performances supérieures aux systèmes individuels. Je présenterai également une stratégie concernant la détection de l'information nouvelle (tâche pilote de DUC'07) avec une approche simple de maximisation/minimisation de cosinus, qui s'est révélée être très performante.

L'ensemble des travaux sur le résumé personnalisé a été réalisé, d'abord dans le cadre du Master recherche de Florian Boudin et puis dans sa thèse de doctorat, qui a été partiellement financée grâce aux FUNDP[1] (Belgique). Les résultats ont étés publiés dans les conférences DUC'06 (Favre et al., 2006) et '07 (Boudin et al., 2007) aux USA, CICling'07 à Mexico (Boudin et Torres-Moreno, 2007a) et RANLP'07 en Bulgarie (Boudin et Torres-Moreno, 2007b).

[1] http://www.fundp.ac.be

6.1 Etat de l'art

J'ai présenté les systèmes de résumé automatique par extraction de phrases au chapitre précédent. Les systèmes de résumé peuvent aussi être divisés dans deux catégories : systèmes de résumé mono-document et multi-documents. Ces derniers peuvent être vus comme une fusion de sorties des systèmes mono-document. Les systèmes multi-documents agissant sur plusieurs textes ont une probabilité plus grande de présenter une information redondante et/ou contradictoire. Des travaux comparant les techniques d'anti-redondance (Newman et al., 2004) montrent qu'une mesure de similarité de type cosine entre phrases (Van Rijsbergen, 1979) a des performances semblables à d'autres méthodes plus complexes telles que *LSI* (Deerwester et al., 1990). Pour l'élimination de la redondance, les recherches se sont focalisées sur la temporalité des documents. Une méthode générale pour aborder les résumés basés sur la nouveauté, consiste à extraire les étiquettes temporelles (Mani et Wilson, 2000) (dates, périodes écoulées, expressions temporelles,...) ou de construire automatiquement une chronologie à partir des documents (Swan et Allan, 2000). Une dernière technique qui utilise la mesure bien connue de χ^2 (Manning et Schütze, 1999) est employée pour extraire des mots et des phrases peu communes à partir des documents.

Les *Document Understanding Conferences* (DUC)

Les premiers systèmes de résumé automatique multi-documents ont été développés dans les années 90 (McKeown et Radev, 1995). La plupart des travaux sur le résumé automatique appliquent des techniques statistiques aux unités linguistiques, telles que les termes, les phrases, etc. pour choisir, évaluer, classer et assembler ces unités selon leur pertinence (Mani et Mayburi, 1999).

Les conférences DUC portant sur la tâche de résumé automatique sont organisées depuis 2001 par le *National Institute of Standards and Technology*[2] (NIST). La tâche principale de DUC consiste à traiter des questions complexes et réelles. Le type de réponse attendue ne peut pas être une entité simple (un nom, une date ou une quantité telle que classiquement défini dans les conférences TREC Question-Answering[3]). Le problème peut se poser comme ceci : étant donnée une thématique et un ensemble $\mathcal{L}$ avec D documents pertinents, la tâche consiste à générer un court résumé de 250 mots, cohérent et bien organisé et qui répondra à/aux questions de la thématique. Les thématiques sont composées de deux parties : le titre et une partie narrative (contenant les questions). Pour les conférences DUC les $D = 25$ documents proviennent du corpus AQUAINT : articles d'*Associated Press*, *New York Times* (1998-2000) et *Xinhua News Agency* (1996-2000)[4].

Comme on avait expliqué au chapitre 5, l'évaluation de la qualité des résumés

[2] http://www-nlpir.nist.gov/projects/duc

[3] http://trec.nist.gov/data/qamain.html

[4] Récemment la conférence DUC'07 a introduit une tâche plus complexe : l'évaluation de systèmes de résumé multi-documents, avec la détection de la nouveauté. Cette tâche sera abordée dans la section 6.3.

(mono-document) reste une tâche pas évidente. En multi-documents le problème n'est pas plus simple. Lors des conférences DUC, des approches manuelles et semi-automatiques ont été utilisées à cette fin. Ainsi *Pyramid* (Passonneau et al., 2005) et *Basic Elements* (BE) (Hovy et al., 2005) ont été employées. Les mesures ROUGE (Lin, 2004) ont été retenues comme des mesures d'évaluation semi-automatiques. Plusieurs mesures manuelles ont été évaluées : cohérence, grammaticalité, non-redondance, pertinence au sujet, qualité linguistique parmi d'autres.

6.2 Neo-Cortex

Notre approche génère un système de résumé multi-documents indépendant de la thématique et basé sur des traitements statistiques. Cortex (Torres-Moreno et al., 2001, 2002) est un système de résumé automatique mono-document présenté au chapitre 5. L'objectif était d'adapter ce système comme un système de résumé multi-documents guidé par une thématique fixée par l'utilisateur. Afin d'obtenir un résumé cohérent à partir de plusiseurs documents, nous avons introduit deux nouveaux paramètres dans Cortex : un paramètre global, la similarité entre un document et la thématique et un paramètre local, le recouvrement de mots entre une phrase et la thématique.

Similarité. Les scores des phrases sont calculés par Cortex sur un seul document. Ils doivent donc être normalisés afin de mettre en évidence le degré de pertinence des documents par rapport à la thématique. En effet, la phrase pertinente d'un document peut avoir un score inférieur à la phrase non pertinente d'un autre document. Ceci est dû à l'indépendance inter-document des scores calculés par Cortex. Le paramètre de similarité (6.1) calculé par un cosinus (Salton, 1989) permet de mesurer la proximité entre deux vecteurs.

L'ensemble de documents est représenté par des vecteurs $\vec{v_d} = (v^1, v^2, \cdots, v^N)$, $d = 1 \cdots Nb_{doc}$; Nb_{doc} étant le nombre total de documents, et la thématique est représentée par le vecteur $\vec{\omega_t} = (\omega^1, \omega^2, \cdots, \omega^n)$, $t = 1 \cdots \tau$; où τ est le nombre total thématiques. N correspond à la taille du vocabulaire (documents et thématique). La similarité est alors calculée par :

$$Sim(\vec{v_d}, \vec{\omega_t}) = \frac{\sum \vec{v_d}.\vec{\omega_t}}{\sqrt{\sum \vec{v_d}^2 + \sum \vec{\omega_t}^2}} \tag{6.1}$$

Le poids $tf.idf$ d'un terme (Salton et McGill, 1983) est une mesure statistique utilisée pour évaluer l'importance d'un terme dans un document. Cette importance augmente proportionnellement par rapport au nombre de fois où le terme apparaît dans le document, mais elle est compensée par l'apparition du terme dans les documents de la collection. Les poids idf ont été calculés sur l'ensemble de documents de la collection DUC :

$$tf.idf_{\vec{v_d},j} = tf_{\vec{v_d},j} \times \log\left(\frac{Nb_{doc}}{n_j}\right) \tag{6.2}$$

$tf_{\vec{v_d},j}$ est la fréquence du terme j dans le document $\vec{v_d}$, n_j est le nombre de documents dans lesquel le terme j est présent. La similarité est normalisée entre $[0, 1]$.

Recouvrement. L'idée est de supposer que les phrases sélectionnées pour constituer un résumé doivent partager une certaine quantité de l'information avec la thématique. Pour estimer cette quantité, nous avons calculé le nombre de mots communs entre la thématique et les phrases S^μ. Le recouvrement R, calculé pour chaque phrase, est la cardinalité normalisée entre $[0, 1]$ de l'intersection entre l'ensemble de mots de la phrase μ et l'ensemble de mots de la thématique T

$$R(S^\mu, \vec{\omega}_t) = \frac{card\{S^\mu \bigcap T\}}{card\{T\}} \tag{6.3}$$

$card\{\bullet\}$ est la cardinalité de l'ensemble $\{\bullet\}$. $\mu = 1 \cdots P$, étant P le nombre total de phrases. Cette mesure génère un meilleur classement des phrases contenant des mots de la thématique et diminue le problème de la phrase hautement classées ne contenant aucun terme de la thématique.

Classement final des phrases. La similarité et le recouvrement sont utilisés pour raffiner les scores de Cortex. Le score final de la phrase s d'un document $\vec{v_d}$ et d'une thématique $\vec{\omega}_t$ est la combinaison linéaire :

$$Score = \alpha_0 \text{Cortex}(s, \vec{v_d}) + \alpha_1 R(s, \vec{\omega}_t) + \alpha_2 Sim(\vec{v_d}, \vec{\omega}_t) \,;\, \sum_i \alpha_i = 1 \tag{6.4}$$

Les valeurs des paramètres α_i sont des poids empiriques. Nous avons appelé $Cortex(\bullet)$ les scores obtenus avec l'équation de l'algorithme de décision, le score $Sim(\bullet)$ calculés avec (6.1) et le score $R(\bullet)$ calculé avec (6.3). Le résumé est généré avec les Λ phrases de plus haut score. Λ, fixé par l'utilisateur, peut être un rapport avec la taille initiale de documents ou à un nombre fixe de phrases. Neo-Cortex combine les paramètres similarité et recouvrement avec la sortie du système Cortex (voir figure 6.1).

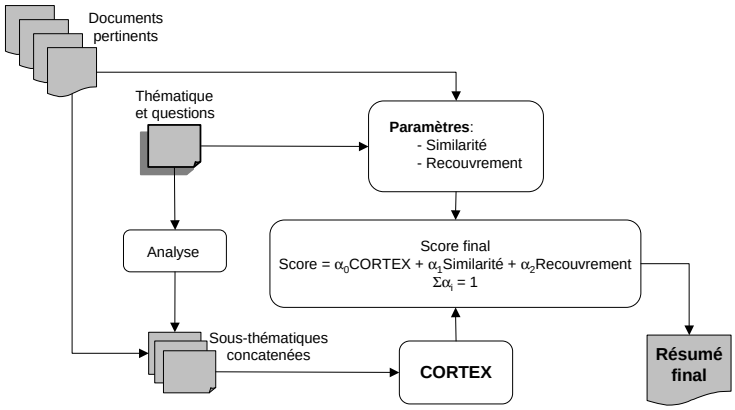


FIG. 6.1: *Architecture de Neo-Cortex*

6.2.1 Adaptations pour la tâche DUC

Thématiques. Nous avons fait une analyse simple de chaque thématique pour créer ζ sous-thématiques (voir un exemple sur la figure 6.2), générées à partir de la concaténation du titre et les questions de la partie narrative.

Numéro et Titre	**Question(s)**
D0603C *Wetlands value and protection*	*Why are wetlands important ?* *Where are they threatened ?* *What steps are being taken to preserve them ?* *What frustrations and setbacks have there been ?*
Sous-thématique 1 : *wetland value protection important* **Sous-thématique 2 :** *wetland value protection threat* **Sous-thématique 3 :** *wetland value protection step preserve* **Sous-thématique 4 :** *wetland value protection frustration setback*	
D0606F *Impacts of global climate change*	*What are the most significant impacts said to result from global climate change ?*
Sous-thématique 1 : *impact global climate change significant*	

FIG. 6.2: *Exemples des thématiques et sous-thématiques pour DUC'06 (D0603C, D0606F) (les sous-thématiques ont été filtrées et lemmatisées).*

Réglage des paramètres. ROUGE (Lin, 2004) a été utilisée pour régler les paramètres de Neo-Cortex (ROUGE-2 et SU4). Nous avons utilisé l'ensemble des données DUC'05 afin d'optimiser les paramètres α_i. La répartition optimale du recouvrement dans le score final de la phrase a été evalué de la façon suivante : nous avons fixé la similarité à 0 et nous avons réalisé un balayage précis (pas des itérations de 0,05) en augmentant le recouvrement jusqu'à ce que nous ayons obtenu la valeur qui optimise les scores ROUGE. Ainsi le meilleur score ROUGE-2 est obtenu avec $\alpha_1 \approx 0,4$ (voir la figure 6.3 à gauche). Les résumés finaux sont obtenus avec un léger post-traitement.

Le paramètre optimal de similarité α_2 est obtenu d'une façon semblable. Le poids α_0 de Cortex et le recouvrement α_1 sont fixés aux valeurs optimales précédentes ($\alpha_0 = 0,6$ et $\alpha_1 = 0,4$). La figure 6.3 à droite, montre deux pics (valeurs maximales pour α_2). Étant donné que la collection de documents DUC'05 n'est pas assez grande et afin d'éviter des erreurs dues à la particularité d'un corpus, nous avons empiriquement choisi le premier pic, $\alpha_2 = 0,11$ (voir la figure 6.3 à droite) donnant le meilleur score. Plusieurs tests ont montré que le recouvrement est un paramètre plus important que la similarité. C'est pourquoi nous avons en premier réglé le paramètre de recouvrement. Les valeurs α_i ont été normalisées afin de fixer les valeurs optimales des paramètres pour DUC'05 : α_0 (Cortex) $= 0,54$ $(0,6 \rightarrow 0,54)$, α_1 (recouvrement) $= 0,36$ $(0,4 \rightarrow 0,36)$ et α_2 (Similarité) $= 0,10$ $(0,11 \rightarrow 0,10)$. D'autres tests ont confirmé que les paramètres trouvés sont optimaux.

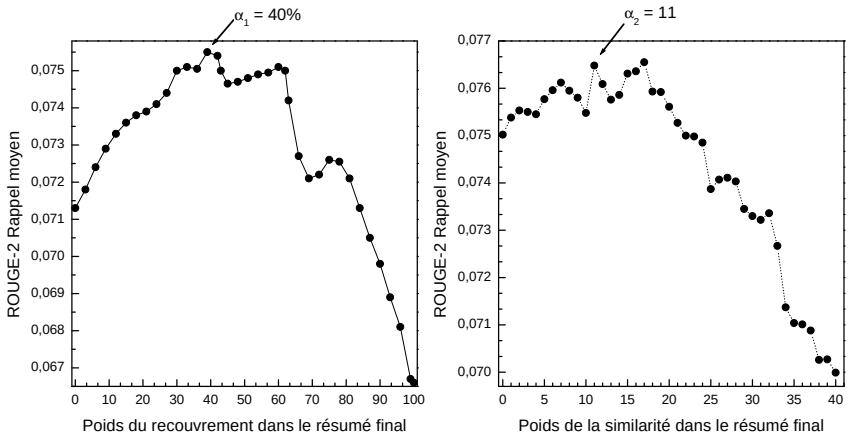


FIG. 6.3: *À gauche, scores de rappel* ROUGE-2 *selon le pourcentage de recouvrement* α_1. *Le facteur de similarité* $\alpha_2 = 0$ *et le facteur Cortex* $\alpha_0 = 1 - \alpha_1$ *(recouvrement). Le score optimal est obtenu avec* $\alpha_1 \approx 40\%$. *À droite, scores de rappel* ROUGE-2 *selon le pourcentage de similarité* α_2. *Le facteur de similarité* $\alpha_2 = 1 - (\alpha_0(Cortex) + \alpha_1(recouvrement))$. *Le score optimal est* $\alpha_2 \approx 11\%$.

Etude des métriques. Le système Cortex peut utiliser plusieurs métriques pour évaluer la pertinence des phrases (Torres-Moreno et al., 2001, 2002). Nous avons examiné empiriquement un éventail de combinaisons (figure 6.4) et choisi finalement trois métriques qui maximisent les mesures ROUGE :

L'angle entre un titre et une phrase (A) : cosinus du produit scalaire normalisé entre la phrase et le vecteur de la thématique.

Deux métriques de Cortex utilisent la matrice de Hamming H, une matrice carrée de $N_L \times N_L$ où chaque case $H[i,j]$ représente le nombre de phrases utilisant exclusivement un des termes i ou j (c.f. section 5.3) : i/ Le poids de Hamming lourd (L) et ii/ la somme de poids de Hamming de mots par la fréquence (O).

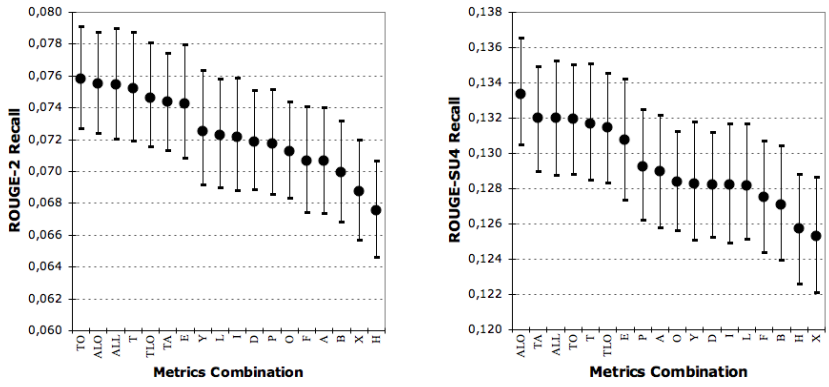


FIG. 6.4: *Scores* ROUGE *de Neo-Cortex pour la tâche de DUC'05 selon la combinaison de métriques utilisées. La combinaison ALL signifie toutes les métriques de Cortex.*

Longueur des phrases. Le système Cortex est incapable de choisir entre deux phrases de même score mais avec des longueurs différentes. Est-ce que les phrases de $10n$ mots sont plus importantes que celles de n mots pour générer un résumé court ? Nous avons réalisé un lissage gaussien des scores de Cortex selon la longueur de la phrase. Des tests complémentaires ont montré qu'un lissage sigmoïdal à la place du gaussien améliore de manière significative les scores ROUGE.

On a comparé les performances globales de Neo-Cortex et de Cortex avec les sept meilleures scores ROUGE des combinaisons de métriques sur l'ensemble de données DUC'05. Les scores ROUGE de la combinaison de toutes les métriques sont améliorés (Boudin et Torres-Moreno, 2007a).

Le système Neo-Cortex a été également comparé aux autres participants de l'évaluation de DUC'05 (voir la figure 6.5). Notre système réalise de très bonnes performances (le meilleur système de tous les scores de ROUGE). Le fait est que l'ensemble de données d'entraînement utilisé pour Neo-Cortex était l'ensemble de données de DUC'05. Neo-Cortex est réglé de façon optimale pour cette évaluation, ce qui explique pourquoi il est très performant.

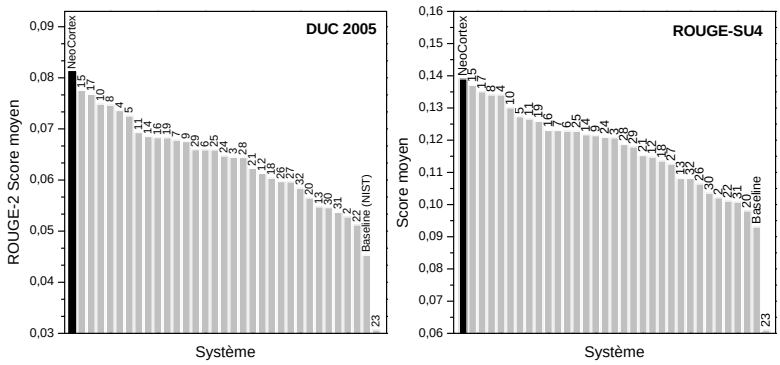


FIG. 6.5: *DUC'05 : scores de rappel* ROUGE-2 *(à gauche) et SU4 (à droite) de Neo-Cortex vs tous les systèmes participants.*

6.2.2 Le système LIA-Thales

Nous avons decidé de participer à la conférence DUC'06. Ceci a été possible grâce à une collaboration entre Thales (Laboratoire MMP), le laboratoire RALI[5] de l'Université de Montréal et le LIA. L'idée principale du résumeur LIA-Thales a été de combiner plusieurs systèmes de résumé. Ce dernier assemble les sorties des systèmes en respectant les contraintes imposées par DUC. Le système LIA-Thales est montré sur la figure 6.6. Les détails des différents modules peuvent être trouvés dans l'article (Favre et al., 2006).

[5] http://rali.iro.umontreal.ca

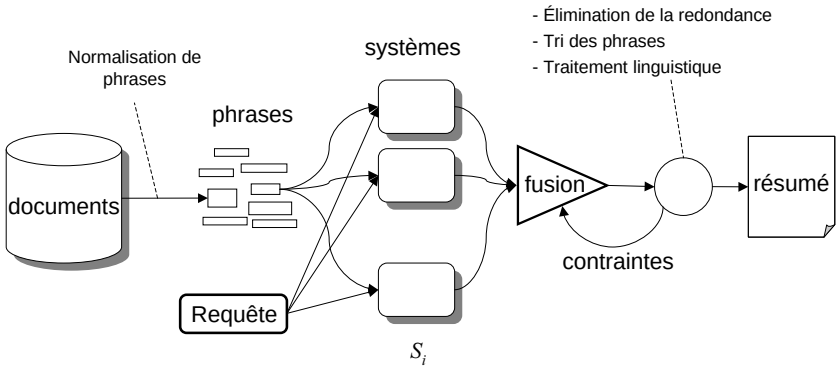


FIG. 6.6: *Architecture principale de la fusion des multiples systèmes pour la sélection des phrases.*

Le pré-traitement inclut la *tokenization*, la normalisation des mots, la recapitalisation des noms propres et l'élimination de mots fonctionnels. Les phrases sont utilisées comme une entrée commune des différents systèmes. La segmentation des textes en phrases considère la structure du document pour détecter les frontières de phrases. La seléction des phrases est principalement basée sur les modèles de résumé non supervisé et la recherche d'information.

DUC'06. Nous avons utilisé quatre systèmes basés sur divers modèles :

(S1) MMR-LSA : Maximal Marginal Relevance (Carbonell et Goldstein, 1998) utilise la similarité entre les phrases dans un espace du type LSA (matrice de co-occurrences réduite).

(S2) Neo-Cortex : nous avons reglé les paramètres de similarité et couverture comme décrit dans la section 6.2.

(S3) Modèle de n-termes de longueur variable. Il utilise les mots de la thématique, les lemmes, les stemes et l'alignement des phrases pour calculer un taux de couverture.

(S7) Score par compacité. Il a été développé pour le module d'extraction des réponses du système de Questions-Réponses de LIA (Gillard et al., 2006). L'idée principale est que la densité et la proximité des mots importants trouvés dans une question aident à extraire la meilleure réponse candidate. Ceci permet de pondérer les phrases en utilisant la densité (« compacité ») et proximité des mots importants de la thématique à l'intérieur de la phrase.

Pour la fusion un graphe de phrases a été construit. Les phrases sont pondérées selon les scores calculées par chaque système de résumé. Des heuristiques simples ont été intégrées pour résoudre les anaphores simples (pronoms et temps). En post-traitment, une réécriture de noms de personnes et des acronymes est effectuée. La première occurrence des acronymes et des noms de personnes utilise les formes complètes, mais les occurrences suivantes sont remplacées par des formes raccourcies. Le post-traitement inclut l'élimination de la redondance en utilisant une technique simple : les phrases qui n'apportent pas suffisament de mots nouveaux sont eliminées. Les phrases qui contiennent de longues formes d'acronymes ou de noms de personnes sont également eliminées.

Nous comparé les performances de Neo-Cortex aux participants de l'évaluation DUC'06. Il obtient de bonnes scores (voir figure 6.7) dans les évaluations semi-automatiques.

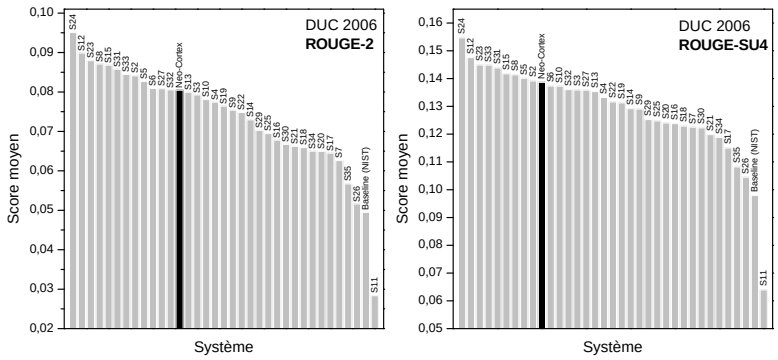


FIG. 6.7: *DUC'06 : scores* ROUGE-2 *(à gauche) et SU4 (à droite) de Neo-Cortex vs tous les systèmes participants. Neo-Cortex est classé 13^{eme} pour* ROUGE-2 *et 10^{eme} pour SU4 sur 35 systèmes.*

DUC'07. Nous avons gardé la même approche que pour DUC'06 en ajoutant d'autres systèmes de résumé :

(S4) Vector Space Model (Buckley et al., 1995) : la similarité entre une phrase et la thématique est calculée en utilisant la métrique LNU*LTC.

(S5) Similarité d'Okapi (Robertson et al., 1996).

(S6) Similarité de Prosit (Amati et Van Rijsbergen, 2002).

Les systèmes S1, S2, S3 et S7 sont très semblables à ceux utilisés en 2006. S4, S5 et S6 sont des implémentations rapides des modèles de récupération (Savoy et Abdou, 2006) pour assurer la diversité dans le processus de fusion.

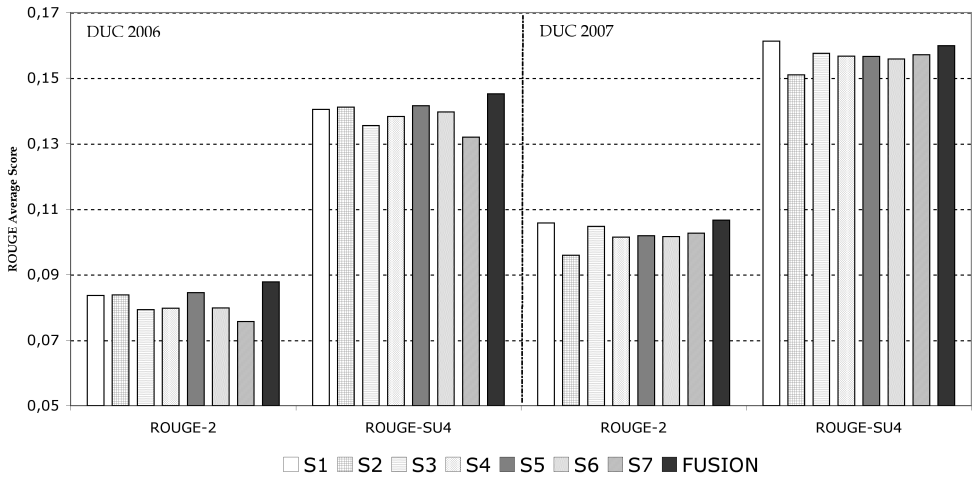


FIG. 6.8: *Rappel* ROUGE-2 *et SU4 des 7 systèmes et la fusion sur les corpus de DUC'06 et '07*

Résultats. La figure 6.8 montre les scores ROUGE obtenus pour nos 7 systèmes sur les corpus de DUC'06 et '07. La fusion des systèmes est également montrée. Ces

résultats corroborent le fait que la combinaison de plusieurs systèmes surpasse le meilleur système et évite le sur-apprentissage. En d'autres termes, assembler des algorithmes de selection des phrases très différents est une bonne stratégie. En effet, la fiabilité de nos systèmes est basse. On observe que Cortex était très performant dans DUC'06 mais dans DUC'07 était le plus mauvais système (dû aux paramètres incorrects). La stratégie de fusion permet surmonter ce genre des problèmes de stabilité.

Je présente les résultats obtenus par notre système (numéro 3) dans DUC'07. Parmi les 30 participants, notre système est classé 9^{eme} dans ROUGE-2 et 11^{eme} dans l'évaluation Basic Elements, 8^{eme} dans SU4 et 8^{eme} dans l'évaluation manuelle. La figure 6.9 montre la position de notre système dans les évaluations automatiques de ROUGE en comparaison avec les autres 29 participants et les deux baselines (numéros 1 et 2). Pour ROUGE-2, notre système a obtenu 0, 106 et pour SU4 0, 159.

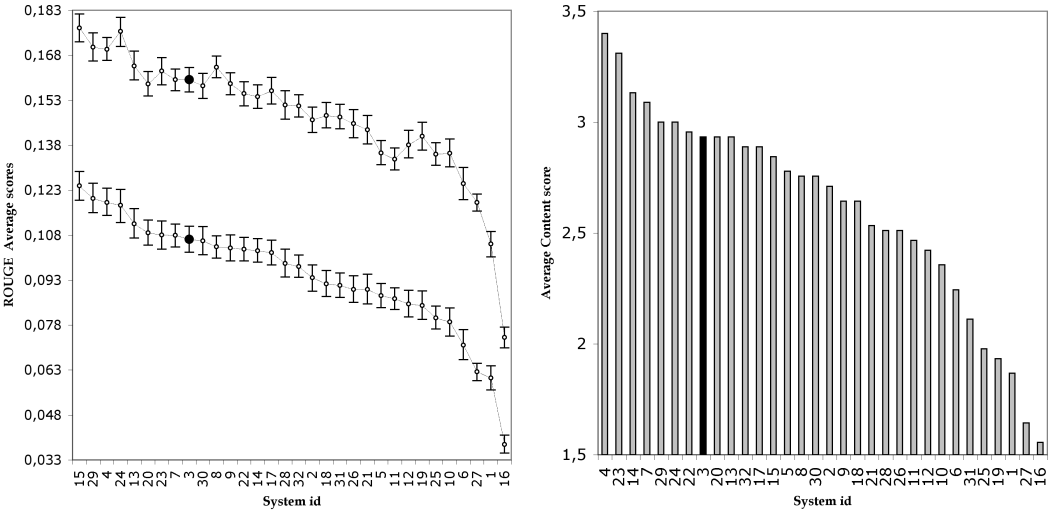


FIG. 6.9: ROUGE-2 *et SU4 pour le 32 systèmes à DUC'07 à gauche. Notre système identifié par le numéro* 3 *(marqué dans la figure par un carré noir), les systèmes* 1 *et* 2 *correspondent à deux baselines. Score moyen de contenu sensible des 32 systèmes à droite. Notre système est identifié avec le numéro* 3 *(marqué dans la figure par la barre foncée).*

En termes de qualité linguistique, le système LIA-Thales présente de bonnes performances, même si nous n'avons pas modifié le traitement linguistique. Les résultats de 2007 (présentés au tableau 6.1) sont semblables aux résultats de 2006, avec un score mauvais dans la non-redondance linguistique (utilisation des pronoms et diversité de formes) et un score moyen dans les autres évaluations. Le problème dans la non-redondance est caractéristique du processus de vote, qui choisit les phrases fortement pertinentes et qui contiennent habituellement les entités nommées les plus fréquentes. Une étape de post-traitement pour détecter et reformuler ces entités pourrait améliorer ces scores.

Mesure	2006	2007
Qualité linguistique moyenne	3.57	3.42
Grammaticalité	4.08	4.11
Non-Redondance	3.84	3.62
Clarté de référence	3.42	3.36
Focus	3.74	3.56
Structure et cohérence	2.76	2.47

TAB. *6.1:* *Scores de qualité linguistique de notre soumission en 2006 et 2007. Il n'y a aucune différence évidente entre les deux évaluations : le module de post-traitement linguistique est le même.*

6.3 Faire simple et beau : la tâche pilote DUC'07

La tâche pilote de DUC'07 demande de produire des résumés multi-documents courts (∼100 mots) et mis à jour, selon une thématique particulière, sur la supposition que l'utilisateur a lu une partie des articles auparavant. Le but de chaque nouveau résumé sera d'informer le lecteur des nouvelles informations. La tâche peut être définie ainsi : étant donné une thématique et trois groupes de documents A, B et C, il faut créer des résumés fluides tels que :

- Un résumé R_A issu des documents de l'ensemble A.
- Un résumé actualisé R_B des documents de B, supposant que le lecteur a déjà lu les documents de A.
- Un résumé actualisé R_C des documents de C, supposant que le lecteur a déjà lu les documents de A et de B.

Les regroupements (*clusters*) de documents doivent être traités par ordre chronologique ; c'est-à-dire, on ne peut pas acceder aux documents du groupe B ou C pour produire le résumé du groupe A ; ni aux documents de C pour produit le résumé de B. Cependant, les documents dans un groupe peuvent être traités dans n'importe quel ordre.

Je présenterai un système de résumé automatique multi-documents, guidé par une thématique avec détection de la nouveauté. Le système sera basé sur une approche de maximisation-minimisation qui repose sur deux concepts principaux. Le premier est l'élimination de la redondance des phrases, qui limite l'information répétitive entre le résumé produit et les précédents. Le deuxième concept est la détection de l'information nouvelle dans un corpus. Nous avons adapté la technique d'extraction (en sac de mots) avec une méthode d'enrichissement qui élargit la thématique avec des informations uniques. Dans l'évaluation de la tâche pilote de DUC'07, notre système a obtenu de très bons résultats lors des évaluations.

6.3.1 Une approche de Maximisation-Minimisation

La motivation principale de cette approche est de mesurer l'information nouvelle contenue dans un regroupement de documents, étant donnée une thématique et un ensemble de documents « déjà vus », et en même temps de réduire l'information redon-

dante. Les avantages principaux de cette approche sont i/ *aucune* connaissance n'est exigée et ii/ le système reste assez indépendant de la langue. Nous avons realisé une *baseline* ayant comme objectif de produire des résumés guidés thématiquement. Des pré-traitements classiques sont appliqués aux corpora : les phrases sont normalisées, filtrées (les mots fonctionnels ou trop communs sont éliminés) et les mots porteurs de sens sont stemmisés avec l'algorithme de Porter (Porter, 1980). Un espace de termes Ξ N-dimensionnel, où N est le nombre de mots type est ainsi construit. Le système score les phrases en calculant la similarité avec le cosinus de l'angle (Salton, 1989). Plus il est petit, plus la similarité est grande. Elle est représentée dans la figure 6.10, par l'angle θ_t, entre le vecteur de la thématique et celui de la phrase. Dans ce cas, $\vec{s}$ est la représentation vectorielle de la phrase candidate et $\vec{t}$ de la thématique.

Élimination de la redondance.

Les phrases issues de multiples documents sont assemblées pour produire un résumé, mais cela engendre des problèmes de redondance. Dans la pratique les phrases d'un regroupement sont scorées en calculant leur angle par rapport à la thématique ; en conséquence, les phrases avec un score élevé sont syntaxiquement proches. Il faut traiter deux problèmes différents de redondance dans un système de détection de nouveauté : la redondance à l'interieur de chaque résumé et la redondance entre les différents résumés. La première concerne la détection de doublons dans le résumé. Nous avons mesuré la similarité entre les phrases faisant déjà partie du résumé et les candidates. Nous avons éliminé ces dernières si leur similarité est supérieure à un seuil T_o, fixé empiriquement. Le deuxième type de redondance est plus spécifique à la tâche : les résumés sont générés en supposant que d'autres résumés ont été précédemment produits. Par conséquent, ils doivent contenir une information additionnelle à celle de la thématique pour informer le lecteur des nouveaux faits. Formellement, les n_p premiers résumés sont représentés comme un ensemble de vecteurs $\Pi = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_{n_p}\}$ dans l'espace de termes Ξ. Notre méthode pour scorer les phrases calcule le rapport entre les deux angles : la phrase $\vec{s}$ avec la thématique $\vec{t}$ et la phrase avec tous les résumés précédents n_p. La valeur la plus petite $\eta(\vec{s}, \vec{t})$ et la valeur la plus élevée $\phi(\vec{s}, \Pi)$ produit le score le plus grand $R(\bullet)$:

$$R(\vec{s}, \vec{t}, \Pi) = \frac{\eta(\vec{s}, \vec{t})}{\phi(\vec{s}, \Pi) + 1} \tag{6.5}$$

où

$$\eta(s, \vec{t}) = \cos(\vec{s}, \vec{t}); \phi(\vec{s}, \Pi) = \sqrt{\sum_{i=1}^{n_p} \cos(\vec{s}, \vec{p}_i)^2}; 0 \leq \eta(\bullet)\,;\, \phi(\bullet) \leq 1 \tag{6.6}$$

Par conséquent $\max R(s) \Longrightarrow \begin{cases} \max\ \eta(\bullet) \\ \min\ \phi(\bullet) \end{cases}$ La phrase avec le score le plus élevé $\vec{s}$ est la plus pertinente selon la thématique (soit $\eta \rightarrow 1$) et simultanément, la plus différente en vue des résumés précédents Π (c'est-à-dire $\phi \rightarrow 0$).

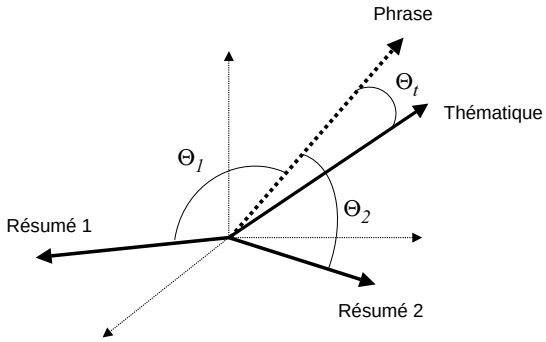


FIG. *6.10:* *Exemple de Maximisation-minimisation de type cosine.*

D'autres travaux (Newman et al., 2004) ont confirmé que la similarité cosine classique est la mesure la plus performante pour éliminer la redondance. Le seuil d'acceptation de la phrase a été reglé empiriquement en utilisant les valeurs de ROUGE comme une mesure de référence. Les scores ROUGE augmentent jusqu'à ce que le seuil atteigne $0,4$ (voir figure 6.11). Une phrase est considérée comme génératrice de redondance si au moins un de ses scores cosinus avec les autres phrases est $> 0,4$.

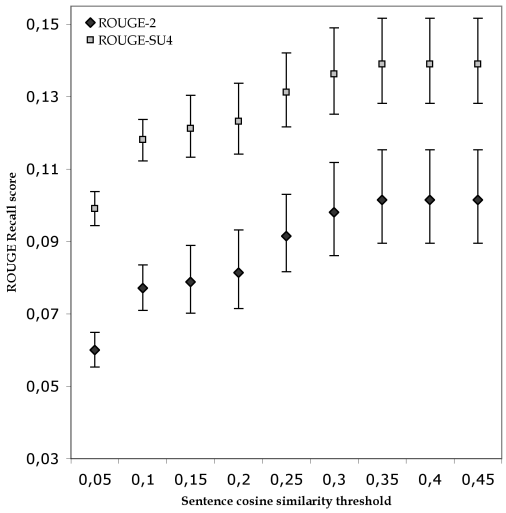


FIG. *6.11:* *Mesure* ROUGE *vs. seuil de la similarité de la redondance.*

Information nouvelle. La détection de la nouveauté est un point critique dans le processus de résumé. En effet, comment détecter, quantifier et utiliser cette information sont des questions difficiles auxquelles nous essayons de répondre avec notre approche. Nous avons utilisé une liste de termes avec les plus haut $tf \times idf$ choisis comme descripteurs de la thématique. Nous avons représenté l'information la plus importante d'un regroupement de documents X par un sac de mots B_X

des n_t termes de plus haut $tf \times idf$. La nouveauté d'un regroupement A par rapport aux regroupements déjà traités est la différence de son sac de mots B_A avec l'intersection de B_A et de tous les sacs de mots des regroupements précédents :

$$B_X = B_X \setminus \bigcup_{i=1}^{i=n_p} B_i \tag{6.7}$$

Le système utilise les termes de B_X pour enrichir la thématique t du regroupement X. La thématique est augmentée par une petite partie de l'information unique contenue dans le regroupement. Les phrases sélectionnées sont guidées par la thématique mais également par l'information unique.

Génération du résumé. Le résumé final est construit par concatenation des phrases avec les scores le plus élevés jusqu'à ce que la taille limite de 100 mots soit atteinte. Par conséquent, la dernière phrase a une probabilité très élevée d'être tronquée. Nous proposons une méthode de la sélectionner afin d'améliorer la qualité de lecture du résumé. Cette méthode est appliquée seulement si le nombre de termes restants est > 5, autrement nous produisons un résumé de taille non-optimale. La phrase après la dernière phrase sélectionée dans la liste de scores est préférée si sa longueur est au moins 33% plus courte et si son score est $> 0,15$. Dans tous les cas, la dernière phrase du résumé est tronquée de 3 mots maximum. On essaie de protéger grammaticalement la phrase en éliminant seulement les mots fonctionnels et les mots à très haute fréquence. Un ensemble ≈ 50 expressions regulières et un anti-dictionaire ont été créés particulièrement pour cette tâche pilote. La figure 6.12 est une vue globale de l'architecture du système.

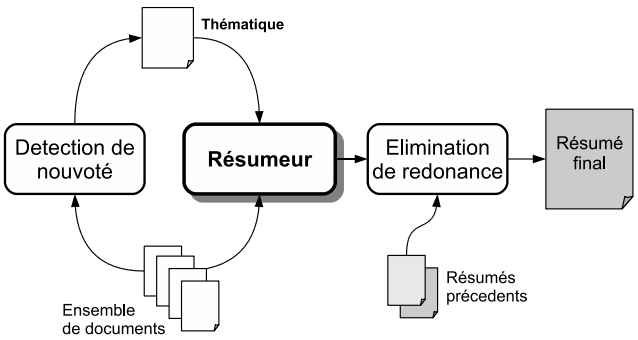


FIG. 6.12: *Architecture générale du système de résumé avec la détection de la nouveauté.*

6.3.2 Expériences

Une des difficultés majeures est d'évaluer et d'optimiser la quantité de termes représentant la « nouveauté » extraits à partir des regroupement. Si trop de termes sont extraits, les résumés produits seront éloignés de la thématique. Inversement, si trop peu de termes sont extraits, la lisibilité des résumés diminuera en raison d'une forte redondance. Nos expériences ont également montré que l'ajout de la nouveauté améliore la

lisibilité et la qualité intrinsèque des résumés produits. L'information contenue dans les résumés est plus hétérogènement repartie, la redondance syntaxique diminue et ainsi la lisibilité et la qualité générale augmentent.

On montre la performance de notre système avec des paramètres optimisées, comparé a celles des 24 participants, dans la tâche DUC'07 (dans laquelle nous avons participé avec une version non-optimale du système, le numéro d'identification du système est 47). On observe sur la figure 6.13 à gauche, que notre système est le deuxième meilleur système dans l'évaluation ROUGE, ceci est une très bonne performance considérant que les post-traitements appliqués sont plutôt génériques. Il y a donc une marge d'amélioration possible dans le post-traitement. Nous étudions actuellement des techniques de réduction des phrases (Master Recherche de Thierry Wasack). Aucun corpus

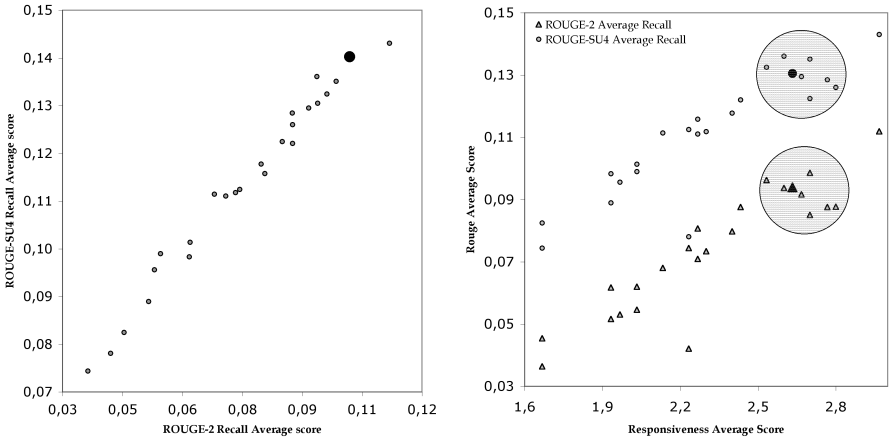


FIG. 6.13: *À gauche : scores* ROUGE-2 *vs. SU4 des 24 participants de DUC'07 dans la tâche pilote (notre système est le cercle gras). À drote :* ROUGE *vs. responsiveness scores. Notre système est le cercle gras pour* ROUGE-2 *et le triangle gras pour SU4.*

d'apprentissage n'était, à l'heure de la soumission, disponible et il n'y avait, à notre connaissance, aucun corpus équivalent pour les systèmes d'apprentissage. Seulement l'évaluation manuelle des sorties des résumés était possible, ceci explique pourquoi les paramètres utilisés pour la soumission du système n'étaient pas optimaux. Les paramètres suivants ont été utilisés pour l'évaluation finale : taille du sac de mots = 15, seuil de redondance = 0, 4, longueur minimale de phrase = 5.

Parmi les 24 participants, notre système a été classé au 4^{eme} rang pour ROUGE-2 et *Basic Elements*, au 5^{eme} dans l'évaluation SU4 et au 7^{eme} dans la évaluation manuelle *responsiveness*. La figure 6.13 à droite, montre la corrélation entre les scores moyens ROUGE-2 et SU4 des systèmes par rapport aux scores moyens *responsiveness*. Le score *responsiveness* obtenu par notre système est de 2, 633, ce qui est au dessus de la moyenne. Notre système se trouve dans le groupe des 8 meilleurs systèmes.

La figure 6.14 à gauche illustre une autre mesure automatique, les *Basic Elements (BE)*. Les scores ont été calculés au moyen de BE-1.1. Notre système obtient BE = 0, 05458, qui est au dessus de la moyenne est positionné 4^{eme} sur 24 systèmes. On voit sur la fi-

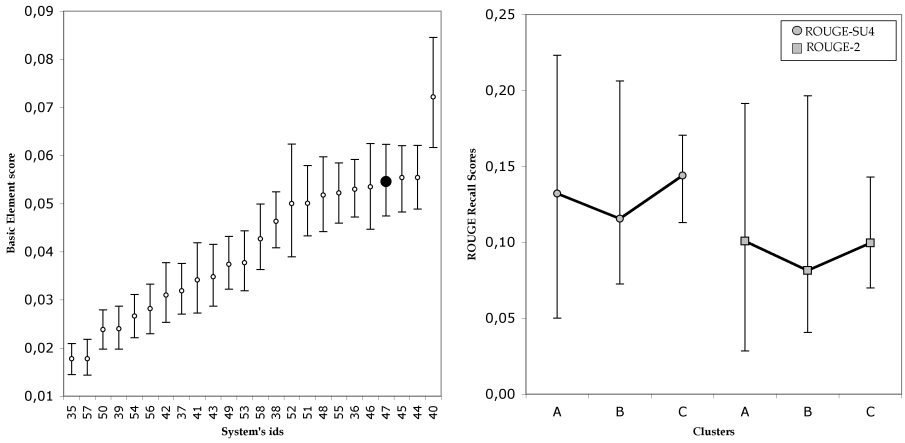


FIG. 6.14: *A gauche : Basic Element (BE) scores des 24 participants de la tâche pilote DUC'07. Notre système est le 47 (marqué par le cercle gras). A droite : scores* ROUGE *de rappel (déviations moyennes maximum-minimum) pour chaque regroupement de documents (articles A∼10, B∼8 et C∼7.*

gure 6.14 à droite, que la moyenne des scores est meilleure pour le dernier résumé (regroupement C) et en plus, les écarts-type diminuent. La stabilité de notre système augmente avec la quantité de documents precedemment traités, la petite perte de performance avec les résumés du regroupement B peut être due à un enrichissement sous-optimal sans un nombre suffisant de termes extraits auparavant.

6.4 Conclusion et travaux futur

La participation à DUC'06 était une excellente occasion d'évaluer la flexibilité du système Cortex sur une tâche nouvelle et différente. J'ai présenté Neo-Cortex, un système de résumé automatique multi-documents basé sur le système Cortex. Nos expériences sur DUC'06 ont montré que Neo-Cortex est un système efficace qui réalise de bonnes performances sur la tâche de résumé multi-documents guidé par une thématique. Le système est cependant, sensible à la segmentation de phrases : les scores ROUGE ont augmenté suivant la qualité de la segmentation. La capacité du système d'être indépendant de la langue est un atout. Dans DUC'06, LIA-Thales, fusion de cinq systèmes de résume, parmi lesquels Neo-Cortex, a obtenu des très bons résultats dans les évaluations automatiques (5ème dans SU4, 6ème dans ROUGE-2, 6ème dans BE et 6ème dans Pyramid) et une bonne performance dans les évaluations humaines (8ème dans le Resp-Overall) (Favre et al., 2006). Il faut approfondir l'étude des combinaisons des métriques afin d'améliorer la qualité des résumés. Nous pensons également utiliser des techniques d'apprentissage afin de trouver automatiquement les paramètres optimaux α_i de la phrase à scorer. Pour DUC'07, nous avons adapté notre approche de 2006 avec de nouveaux systèmes dans le processus de fusion. Les résultats ont confirmé que la fusion apporte plus de stabilité et réduit le risque de sur-apprentissage. Un pe-

tit module de post-traitement basé sur des règles linguistiques simples a amélioré les résultats. Les travaux futurs incluent le paradigme de fusion et l'implémentation de la compression de phrases à la tâche de détection de la nouveauté. D'une manière plus générale, la détection de la nouveauté a besoin d'une évaluation spécifique de la redondance à partir de l'information déjà vue. À long terme, cela ouvre la voie sur l'évaluation du résumé oral (thèse de Benoît Favre), qui est d'un grand intérêt pour le LIA.

Nous avons également participé à la tâche pilote de DUC'07, avec une approche simple qui évite la redondance. Elle sélectionne les phrases proches de la thématique, en négligeant l'information déjà connue. Puis, la nouvelle information est augmentée en ajoutant à la thématique les mots apparaissant seulement dans les nouveaux documents. Ce système est très performant par rapport aux 24 participants. Les résultats de nos expériences précisent plusieurs questions et directions de recherche pour les travaux futurs. La détection de la nouveauté d'information dans les groupes de documents introduit trop de bruit dans les résumés. Si l'on considère seulement les phrases les plus pertinentes pour l'extraction de termes, on devrait augmenter les performances. Des applications dans un domaine spécialisé, la chimie organique (thèse de Florian Boudin), sont actuellement à l'étude. Ce système permettra aux utilisateurs de gagner du temps en ne proposant à lire que les nouveaux faits, en évitant les informations déjà connues.

Chapitre 7

Applications au raffinement de requêtes

Comprendre le sens d'un mot,
c'est savoir quelles phrases il est possible de construire
à partir de lui.
Jean Cohen.

En 2006 Éric SanJuan et moi discutions des autres applications possibles, mis à part le résumé, de l'algorithme Cortex et sur sa possible combinaison avec un système symbolique ou linguistique. Cortex, censé être un extracteur de phrases, pouvait-il jouer un rôle dans cette tâche si éloignée de son domaine ? Soit un corpus de résumés (abstracts d'un journal, par exemple). Chaque abstract peut être vu comme la phrase d'un pseudo-document qui est le corpus en entier. Cortex pourrait être donc appliqué à extraire des *phrases* (donc des résumés) du corpus afin d'en trouver les plus pertinentes... les plus pertinentes par rapport à quoi ? à la *requête* d'un utilisateur évidemment. Dans ce chapitre, nous visons le classement de documents dans un domaine fortement technique dans le but de rapprocher ce classement à celui obtenu par une ontologie existante (structure de connaissances). Nous avons testé et combiné des modèles symboliques et vectoriels. L'approche symbolique s'appuie sur une analyse peu profonde et des relations linguistiques internes entre termes à plusieurs mots. L'approche vectorielle consiste à classer les documents avec différentes fonctions de classement s'étendant du tf.idf classique jusqu'aux fonctions de similarité plus élaborées du résumé automatique Cortex (c.f. chapitre 5). Les résultats montrent que le classement obtenu par l'approche symbolique est plus performant que le modèle vectoriel sur la plupart des requêtes. Cependant, le classement obtenu en combinant les deux approches surpasse largement les résultats obtenus séparément par les deux approches. L'ensemble des résultats de cette étude, réalisée conjointement avec Fidelia Ibekewe, Éric SanJuan et Patricia Velázquez a été publié dans le congrès *Applications of Natural Language to Data Bases*, NLDB'07 (SanJuan et al., 2007).

7.1 Introduction

En dépit de l'énorme quantité d'études portant sur l'expansion de requêtes et la classification documentaire, ce sujet continue à attirer beaucoup d'attention. En effet, des études précédentes ont établi que les utilisateurs utilisent rarement les options de recherche avancée disponibles sur la plupart des moteurs de recherche ou dans les bases de données spécialisées. La longueur moyenne d'une requête est $\approx$ 1,8 mots (Ray et al., 1997). Ceci signifie que les termes de la requête sont souvent trop imprécis. Dans les domaines techniques, on peut s'attendre à ce qu'une catégorie sémantique unique puisse être associée à chaque terme du domaine (une phrase nominale qui se réfère à un concept unique d'un domaine spécialisé). Quand une ontologie existe, le raffinement par des termes voisins sémantiquement proches consiste en une expansion des termes de la requête en utilisant les termes de sa même catégorie. Quand la requête est trop imprécise, ce processus de raffinement par les termes contigus sémantiquement proches permet de classer les documents. Ce classement est fait selon la fréquence de ces termes dans les titres ou les résumés disponibles des bases de données bibliographiques. Nous visons le classement de documents dans un domaine technique afin de le rapprocher au classement obtenu par une ontologie. Le classement de référence est obtenu en raffinant les termes de la requête avec les termes dans la même catégorie sémantique dans l'ontologie. Dans ce contexte, un pré-requis est que les termes du domaine dans le corpus de test soient précédemment annotés et assignés à une catégorie sémantique unique dans l'ontologie. Nous avons examiné deux approches de classement, les méthodes symboliques et les modèles vectoriels, que nous essayerons avec l'objectif d'obtenir les classements les plus proches possibles du classement de référence mais sans employer les termes manuellement annotés ni la catégorie sémantique d'un terme dans l'ontologie.

Nous explorons les deux approches principales pour le raffinement de requêtes : l'approche du modèle vectoriel qui mesure la similarité *termes-document* et une approche symbolique basée sur les relations linguistiques extérieures entre les termes de la requête et les documents. Nous avons implanté le modèle vectoriel en utilisant le système Cortex, initialement conçu pour le résumé automatique (Torres-Moreno et al., 2001, 2002). L'approche symbolique du système TermWatch (SanJuan et Ibekwe-SanJuan, 2006) extrait les termes multi-mots, les lie par des relations morphologiques locales, lexicales, syntaxiques et sémantiques et regroupe les variantes de ces termes en considérant ces relations. Étant donné un terme de la requête, ces regroupements sont utilisés pour classer les documents selon la proportion des termes partagés entre les regroupements et les documents qui contiennent également le terme de la requête. L'idée est de raffiner un terme de la requête avec les termes voisins sémantiquement les plus proches (*semantic nearest neighbour* (S-NN)). Finalement, dans une approche hybride, les relations de classement utilisées dans l'approche symbolique sont combinées aux différentes fonctions du modèle vectoriel afin de voir si ceci améliore les résultats obtenus par chaque modèle séparément. Ces méthodes ont été évaluées contre un classement de référence obtenu par classification de documents en utilisant des catégories sémantiques à partir d'une taxonomie construite à la main associée au corpus de test. Comme

un sous-produit, cette expérience fournit également une nouvelle méthodologie pour comparer les méthodes issues de deux approches complètement différentes.

7.2 Corpus de test

Afin d'avoir un classement de référence, nous avons besoin d'un corpus avec une structure de connaissance associée (taxonomie ou ontologie) où chaque terme pourra être dépisté facilement. Nos systèmes de raffinement de requêtes extraient les termes automatiquement à partir du corpus et la structure de connaissance associée sera utilisée pour construire le classement de référence. Le corpus GENIA[1] satisfait nos exigences car il comporte une ontologie construite à la main et les termes obtenus des résumés ont été manuellement annotés et assignés aux catégories par des spécialistes du domaine. Ce corpus est composé de 2 000 références bibliographiques extraites de la base MEDLINE[2], en utilisant les mots-clés : *Human, Blood Cells* et *Transcription Factors*. Dorénavant nous nous référerons aux titres et les résumés de ces références comme de documents. Il y a 36 catégories et un total de 31 398 termes. La plus grande catégorie, appelée « *other name* » a 10 505 termes suivie de « *protein molecule* » avec 3 899 termes et « *dna domain or region* » avec 3 677 termes. La distribution des termes dans les catégories obéit à une loi de Zipf. Dans ce contexte, chaque terme annoté peut être vu comme une requête potentielle que sera extraite de tous les documents du corpus contenant ce terme ou des termes sémantiquement proches dans la même catégorie GENIA. Les documents extraits peuvent donc être classés selon le nombre de termes annotés dans la même catégorie GENIA comme la requête de terme. Le classement obtenu pour chaque requête en utilisant les termes manuellement annotés et les catégories GENIA constitue le classement de référence. L'expérience de raffinement de requête consiste à tester la capacité des différentes méthodes pour produire un classement aussi semblable que possible au classement de référence. Naturellement, aucune des méthodes de raffinement de requête examinées n'a utilisé les termes manuellement annotés ni a eu connaissance préalable de leur catégorie sémantique dans l'ontologie GENIA. Les termes des requêtes utilisées dans nos expériences sont de termes manuellement annotés, qui se trouvent dans au moins 50 documents et qui ont été associés à une catégorie autre que « *other name* » dans le corpus GENIA. Nous avons exclu les termes d'un seul mot comme « *cell* » car dans le corpus ce terme existe pratiquement dans tous les documents. Seize requêtes avec des termes multi-mots remplissent ces critères. Le tableau 7.1 montre les termes des requêtes et leur catégorie GENIA, le nombre d'éléments dans cette catégorie et le nombre de documents contenant ce terme.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA
[2] http://medline.cos.com

Requête	Catégorie GENIA	Nb	Docs
activated T cell	cell_type	1723	51
B cell	cell_type	1723	120
Epstein-Barr virus	virus	352	66
glucocorticoid receptor	protein_family_or_group	2452	96
human immunodeficiency virus type 1	virus	352	52
human monocyte	cell_type	1723	69
Jurkat cell	cell_line	1992	66
Jurkat T cell	cell_line	1992	58
NF-kappa B	protein_molecule	3885	271
nuclear extract	cell_component	205	74
nuclear factor	protein_family_or_group	2452	54
nuclear factor of activated T cells	protein_family_or_group	2452	51
protein kinase C	protein_molecule	3885	83
T cell	cell_type	1723	339
T lymphocyte	cell_type	1723	115
transcription factor	protein_family_or_group	2452	487

TAB. 7.1: *Requêtes utilisées dans les experiences.*

7.3 Méthodologie

7.3.1 Approche symbolique

L'approche symbolique du système TermWatch (SanJuan et Ibekwe-SanJuan, 2006), est composé de trois modules : un extracteur de termes, un identificateur de relations (qui produit le réseau terminologique) et un module de regroupement. Le regroupement est basé sur des relations linguistiques générales qui ne dépendent pas d'un domaine particulier et qui ne demandent pas d'adaptations spécifiques si l'on change de corpus. Différentes relations linguistiques pour l'expansion des termes de la requête vers leurs correspondants termes *S*-NN ont été évaluées. On a commencé à classer à partir d'une granularité grossière, telle que le type grammatical des mots d'entête (*head basic clustering*) vers une granularité plus fine. Ainsi, n'importe quel terme de la requête est repéré dans l'ensemble de termes *S*-NN extraits automatiquement. Comme ces termes *S*-NN ont été regroupés, le terme de la requête peut être représenté par le vecteur du regroupement avec autant de dimensions que de regroupements et dont ses valeurs sont le nombre de variantes que la requête possède dans chaque regroupement. Puisque chaque document peut également être représenté par un vecteur similaire qui donne pour chaque regoupement le nombre de ses termes dans le document, la pertinence du document par rapport à la requête peut être évaluée comme le produit scalaire entre les deux vecteurs (regroupement et document).

– **Classement par occurrences des mots d'entête (Head).** Il consiste à classer les documents en s'appuyant sur un calcul d'occurrences des mot d'entête de la requête dans les documents qui contiennent ce mot sans se soucier de leur position grammaticale. La justification d'une utilisation assez commune est le rôle bien connu des noms d'entête dans les phrases nominales : ils évoquent le sujet de la phrase, donc de la requête. De cette manière les documents dans lesquels le mot

d'entête avec une haute fréquence pouvaient correspondre aux documents avec le plus haut nombre de termes dans la même catégorie GENIA.
Le classement de documents avec cette relation est exécutée en dehors de TermWatch puisqu'il s'appuye simplement sur un calcul de l'occurrence d'un mot d'entête dans les documents.

- **Classement par regroupement de base de TermWatch (TW).** La plupart des relations de regroupement à grande échelle dans TermWatch consistent à fusionner tous les termes qui partagent le même mot du titre dans le même regroupement. Cette relation génère des regroupements avec des entêtes identiques. Etant donné un terme de la requête, les documents sont classés selon le nombre de leurs termes qui ont le mot d'entête du terme de la requête également en position de *head*. Par exemple, étant donné la requête *T cell* où *cell* est le mot d'entête, le document classé le plus haut par cette relation a eu le plus grand nombre de termes avec « cell » en sa position d'entête : *B cell, cell, blood cell, differentiated cell, hematopoietic cell, HL60 cell, L cell, lympoid cell, macrophage cell, monocyte-macrophage cell, nucleated cell, peripheral blood cell, S cell, T cell.*
- **Classement par regroupement sémantique sévère (Comp).** Il consiste à classer en utilisant les termes dans les composants connectés constitués par les variantes orthographiques, par de substitutions de variantes synonymes acquises au moyen de WordNet[3] et par les relations d'expansions (où seulement un mot a été ajouté au terme). L'idée est de limiter les S-NN d'un terme de requête à seulement les termes qui ne comportent pas un eloignement thématique et qui sont les plus proches S-NN de toutes les relations possibles utilisées dans TermWatch.
- **Classement par regroupement sémantique laxiste (Var).** Les relations sont ajoutées à celles de *Comp* afin de former les plus grands regroupements qui impliquent les plus faibles variantes d'expansion (addition de plus d'un mot modificateur) et la substitution des mots modificateurs. L'idée est d'augmenter le S-NN d'un terme de requête aux voisins sémantiquement plus lointains où le lien avec le sujet original du terme de requête peut être plus faible.

7.3.2 Approche du modèle vectoriel

Nous avons examiné deux approches de classification de documents basées sur le modèle vectoriel. La première méthode suppose que la fréquence d'un mot peut être estimée dans l'ensemble complet de documents représentés comme un fichier inversé. La deuxième méthode utilise l'ensemble restreint de documents qui contienent au moins une occurrence du terme de la requête.

Soit Δ l'ensemble de tous les résumés dans la base de données bibliographique et Ω l'ensemble d'unitermes (termes avec seulement un mot), pour n'importe quel d résumé, nous dénoterons par Ω_d l'ensemble d'unitermes qui apparaissent au moins une fois dans d et par Δ_w l'ensemble de documents dans lequels w apparaît.

Nous assumons l'existence d'un fichier inversé pour lequel tout mot w appartenant

[3] http://wordnet.princeton.edu

au résumé d dans la base de données bibliographique donne la fréquence $f_{d,W}$ de w dans d. Basé sur un tel fichier inversé, les documents peuvent être classés par rapport au *tf.idf* score des termes de requête dans le document avec ou sans mécanisme d'expansion de requête *QE*. L'approche consiste à calculer la fonction *tf.idf* puis de remplacer le vecteur de termes de la requête par la somme des vecteurs des documents classés en tête. Cet expansion de requête est alors utilisée pour améliorer un autre classement.

Maintenant, nous ne considérons plus l'existence d'un fichier inversé. Etant donné une séquence de requête T sous la forme de MWT les mesures suivantes sont calculées sur l'ensemble restreint de documents $\Delta(T)$ où la chaîne T apparaît. Ces documents sont représentés dans un espace vectoriel (Salton, 1971; Morris et al., 1999) en utilisant le système Cortex (Torres-Moreno et al., 2001, 2002) qui inclut un ensemble de métriques indépendantes combiné avec un Algorithme de Décision. Cette représentation vectorielle considère les noms, les mots composés, les verbes conjugués, les numéros (numériques et/ou textuels) et les symboles. Les autres catégories grammaticales comme les articles, les prépositions, les adjectifs et les adverbes sont éliminées en utilisant un antidictionnaire. Des processus de lemmatisation et de stemming (Paice, 1990b; Porter, 1980) sont effectués pour augmenter les fréquences des mots. Les mots composés sont identifiés, puis transformés en unitermes lemmatisés uniques en utilisant un dictionnaire. Pour décrire les métriques selectionées pour les raffinement de requêtes, nous utiliserons la notation suivante pour n'importe quel $w \in \Omega$ et $d \in \Delta(T)$:

$$\Delta(T)_w = \Delta_w \cap \Delta(T) \qquad f_{d,.} = \sum_{\omega \in \Omega_d} f_{d,\omega} \qquad f_{.,w} = \sum_{\delta \in \Delta(T), w \in \Omega_\delta} f_{\delta,w}$$
$$\Omega(T) = \{\omega \in \Omega : f_{.,w} > 1\} \qquad f_{.,.} = \sum_{\omega \in \Omega(T)} f_{.,\omega} \qquad \Omega(T)_d = \Omega_d \cap \Omega(T)$$

Nous avons testé les métriques décrites au-dessus aussi bien que leurs combinaisons : l'angle (A), trois mesures différentes de recouvrement de la requête (D, L, O) et la fréquence de mots informatifs (F). Nous avons également considéré les combinaisons suivantes de l'ensemble de métriques $\{A, D, O\}$, $\{A, L, O\}$, $\{A, D, L, O\}$, $\{F, L, A, D, O\}$ basé sur l'algorithme de décision Cortex.

A est l'angle entre T et d. Tous les mots dans T n'ont pas la même valeur informative, puisque les mots les plus proches du terme *head* ont une probabilité plus élevée d'avoir une corrélation avec la catégorie des termes. Ainsi, nous avons représenté le terme de la requête $T = t_1 ... t_n h$ par un vecteur $\vec{T} = (x_w)_{w \in \Omega(T)}$ où :

$$x_w = \begin{cases} 15 & \text{si } w = h \\ j & \text{si } w = t_i \text{ pour } i \in [1..n] \\ 0 & \text{autrement} \end{cases}$$

D est la somme des fréquences de mots dans le d résumé multiplié par sa probabilité d'occurrence en $\Delta(T)$ comme suit : $D(d) = \sum_{w \in \Omega(T)_d} \left(\frac{f_{.,w}}{f_{.,.}} \times f_{d,w} \right)$

O Se réfère aux documents impliquant les termes qui ocurrent dans presque tous les documents : $O(d) = \sum_{w \in \Omega(T)_d} (|\Delta(T)_w| \times f_{d,w})$

L indique les documents qui recouvrent avec les mots de la requête mais avec un plus grand vocabulaire : $L(d) = |\Omega(T)_d| \times \sum_{w \in \Omega(T)_d} (|\Delta(T)_w|)$

F est la somme de frequences des termes $F = f(., w)$ Elle favorise les documents avec un petit vocabulaire, à l'opposé des métriques D, O, L.

7.3.3 Approche hybride

Les regroupements construits par TermWatch visent un degré élevé d'homogénéité sémantique. Ils se fondent sur l'existence d'une famille restreinte de relations de variations linguistiques parmi les termes et ainsi elles sont généralement petites. Par conséquent, quand on projete un terme de la requête T sur son S-NN termes dans les regroupements, ceci saisit souvent seulement quelques regroupements. Ainsi, le classement des documents selon leur recouvrement avec ces regroupement produit une proportion d'attaches substantielle. Nous avons alors essayé d'utiliser les métriques normalisées de Cortex pour casser ces attaches. En effet comme précisé dans la séction précédente, les hauts scores des métriques selectionées de Cortex sont obtenus pour les documents contenant les mots de la requête dans T et les mots qui sont fréquemment associés à eux, c'est-à-dire leurs contextes de co-occurrence. Puisque les scores des documents basés sur le recouvrement de regroupements de documents sont des entiers, les *tails* peuvent être simplement cassées en ajoutant à ces scores le score de décision de Cortex qui est un nombre réel entre [0,1]. Ceci mène à un nouveau système de classement de documents (montré sur la figure 7.1) où les documents sont :

- Extraits en mode booléen du texte complet basé sur une phrase exprimé en langage naturel ;
- Rangés selon les relations linguistiques qu'ils partagent avec les termes multimots de la requête ;
- Re-classés par rupture des attaches basés sur le vector de similarités avec la requête.

Nous avons testé si des combinaisons particulières des méthodes à partir des deux approches amélioraient la performance des classements obtenus séparément par chaque méthode. Les résultats obtenus par chaque méthode sont décrits ci-après (voir la section 7.4), nous avons raffiné le classement de TermWatch en utilisant les métriques de Cortex mais nous avons également testé la combinaison des métriques de Cortex avec les regroupements HEAD. Les classements basés sur regroupements de TW et raffiné en utilisant n'importe quelle combinaison $X_1...X_n$ des métriques de Cortex sont dénotées par $X_1...X_n$-tw (respectivement $X_1...X_n$-HEAD).

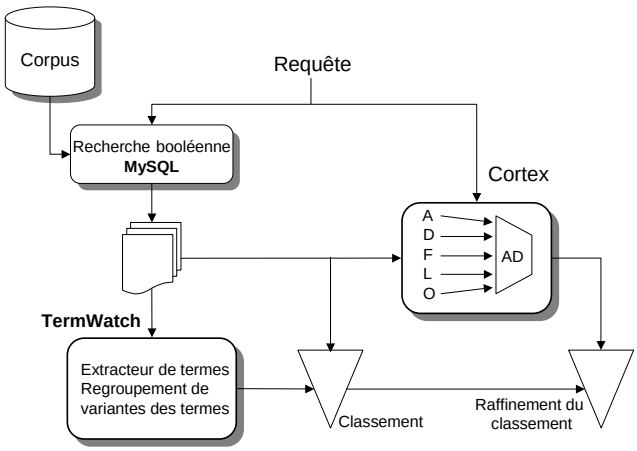


FIG. 7.1: *Système de classement hybride.*

7.4 Résultats

Nous avons analysé les résultats à partir des approches vectorielle, symbolique et hybride. Étant donné un terme de la requête, nous évaluons les méthodes décrites dans les sections 7.3.1 et 7.3.2 selon leur capacité de classement de documents en respectant une ontologie existante, c'est-à-dire, les documents classés en tête devraient contenir les termes de la catégorie sémantique dans l'ontologie GENIA comme le terme de la requête. Pour chaque requête, on a comparé les classement de documents produits par les différentes méthodes au classement de référence, avec le coefficient de concordance W[4] de Kendall (Siegel et Castellan, 1988). Nous avons calculé le coefficient W de Kendall et son « p-value » en utilisant le logiciel **R** pour le calcul statistique avec le paquet Concord[5]. Nous n'avons utilisé ni la précision ni le rappel comme évaluations car toutes les méthodes de classement utilisent la même liste de documents, elles sont donc toutes basées sur la sélection de documents qui contiennent les termes de la requête. La seule différence a été la manière comment les méthodes ont classé les documents. C'est pourquoi les calculs de rappel et de précision n'ont aucun sens ici.

7.4.1 Comparaison globale des méthodes

La figure 7.2 montre les *boxplots* (boîtes à moustaches) du coefficient de concordance W de Kendall sur toutes les requêtes pour chaque méthode. Selon les *boxplots*, le raffinement du classement de TW avec les métriques de Cortex $(X_1, \cdots, X_k)$-TW où

[4]Ce coefficient est issu d'une famille de tests non-paramétriques et robustes qui ne font aucune supposition sur la distribution gaussienne des données. Le coefficient W de Kendall vaut 1 dans le cas d'accord complet entre deux classements et 0 pour un désaccord total. Comme dans tous les tests statistiques, pour interpréter les valeurs intermédiaires, il est nécessaire de vérifier si les scores obtenus par une méthode sont significativement différents de ceux d'un classement aléatoire sur les mêmes données.

[5]`http://www.r-project.org`

$X_1, \cdots, X_k$ et la combinaison de {A, D, F, O, L}) surpasse la performance de TW simple, qui à son tour surpasse la méthode HEAD, toutes les métriques de Cortex (A, D, F, O, L) prises séparément ou combinées et les classement de MySQL (tf.idf et QE). Nous avons vérifié si ces différences sont statistiquement significatives. Nous avons appliqué le test du classement non paramétrique de Wilcoxon et le test du classement total de Friedman, les deux disponibles dans le paquet du logiciel R. Ces deux tests sont utilisés pour comparer les scores moyens de W de Kendall obtenus pour chaque méthode.

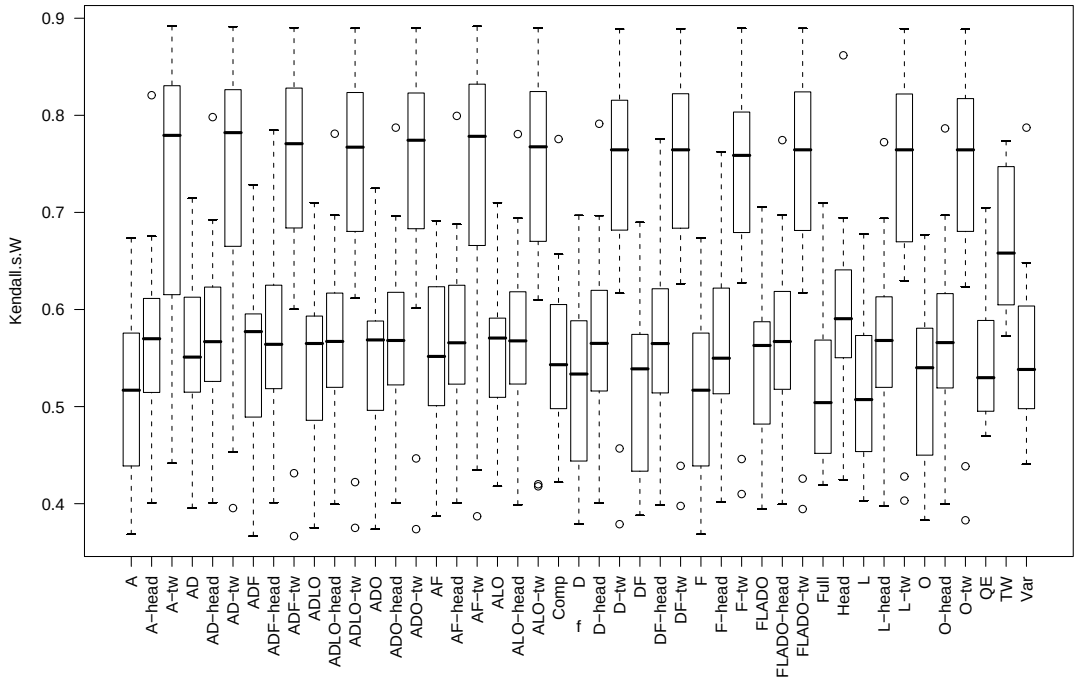


FIG. 7.2: *Les boxplots montrent les scores moyens du W de Kendall et les valeurs extrêmes pour chaque méthode. Les symboles A, D, F, L, O et leurs combinaisons dans la caisse au dessus se réfèrent aux métriques de Cortex (par exemple FLADO) ; « Head », « TW » et « Var » se réfèrent aux classement basés sur les regroupements de TermWatch respectifs. Les symboles représentent les métriques de Cortex suivis par la caisse au dessous de « tw » ou « head » référé aux approches hybrides. « QE » représente tf.idf avec QE. Reproduit de (SanJuan et al., 2007), page 259.*

Premièrement nous avons analysé les combinaisons des métriques de Cortex pour trouver quelle avait les meilleures performances. Le test de Friedman a montré, avec une confiance de 99%, qu'il existe des différences significatives. Cependant, réalisant le même test mais seulement avec la combinaison d'au moins deux mesures de Cortex parmi {A, D, O, L} nous n'avons pas trouvé des différences significatives parmi les membres de ce regroupement (p-value > 0.8). Ceci montre que la combinaison des métriques de Cortex basée sur son algorithme de décision 7.3.2 améliore de manière significative les résultats.

Maintenant, en observant le groupe de méthodes basées sur une simple métrique de Cortex, on voit des différences significatives trouvées par le test de Friedman avec une confiance de 99%. En effet, basé sur le test de Wilcoxon nous avons trouvé que O et D ne sont pas statistiquement différents (p-valeur=0.86), même chose pour F et L (p-valeur=0.82). Les deux premiers semblent être plus adaptés à cette expérience que F et L (voir leurs valeurs de W de Kendall à la figure 2). Les métriques O et D classent au premier les documents dans lesquels les mots fréquents correspondent aux mots de la requête ou sont fortement associés à eux, considérant que les métriques L et F se concentrent sur la couverture de vocabulaire des documents, indépendamment des mots de la requête. L est très sensible aux documents avec une couverture de vocabulaire large et F fait l'inverse. Ainsi ces deux documents basés sur des critères intrinsèques aux documents classés mais pas à la requête. La métrique A qui considère la position de chaque mot dans la requête reste à part. Finalement, nous avons vérifié les performances parmi les méthodes symboliques pour voir s'il y a une différence statistique de leurs classements. Le test de Wilcoxon a permis de vérifier que l'hypothèse des moyennes égales entre le *TW de base* et les classements de *Head* sont rejetées avec un risque plus bas que 5%. Le même test a également montré avec une confiance de 90% que la méthode *Head* surpasse celle de *Var* mais que les differences observées entre les classements *Head* et *COMP* n'étaient pas statistiquement significatives (p-valeur=0.23). Maintenant, nous comparons les classements obtenus par l'approche hybride. Nous avons déjà observé qu'il n'y a aucune différence statistique entre les scores moyens des combinaisons d'au moins deux métriques de Cortex. Nous observons le même phénomène entre n'importe quel classement de TermWatch raffiné avec n'importe quelle métrique de Cortex. En effet, la p-value résultant du test de Friedman sur cette famille des méthodes est supérieur à 0,54.

Puisque nous avons déjà vérifié l'efficacité de l'algorithme de décision de Cortex, nous aurons besoin seulement de considérer *FLADO-tw* qui est le raffinement du classement de TW basé sur la combinaison de toutes les métriques de Cortex sélectionnées parmi toutes les combinaisons possibles. De la même manière, nous avons découvert qu'il n'y a aucune évidence statistique des différences entre les raffinements des classements de HEAD avec n'importe quelle métrique de Cortex. Ainsi nous considérerons seulement la combinaison de *FLADO-Head*. Nous obtenons alors, basés sur le test de Wilcoxon, que *FLADO-tw* surpasse *TW* avec une confiance de 95%, et que *Head* surpasse *FLADO-Head* avec une confiance de 99%. Puisque nous avons précédemment montré que *TW* surpasse *Head*, nous déduisons que *FLADO-tw* clairement surpasse *FLADO-Head* et *FLADO*. Ceci s'est avéré être le cas avec un niveau de confiance plus haut que 99.98%. Après ces tests statistiques, il semblerait qu'utiliser uniquement la combinaison des métriques de Cortex FLADO choisie par son algorithme de décision pour raffiner les classements sémantiques de TW, génère la meilleure approche hybride pour le raffinement de la requête. Contrairement, les résultats considérablement dégrades en raffinant le classement produit par la méthode *Head* avec les métriques de Cortex.

7.4.2 Comparaison des méthodes de classement requête par requête

Les résultats globaux peuvent masquer des différences importantes comme suggéré sur la figure 7.2 en regardant la longueur des boîtes et les valeurs extrêmes. La vue des détails des performances pour les méthodes principales est montrée dans le tableau 7.2. Ce tableau montre les scores du W de Kendall de chaque méthode par requête. Pour chaque requête, seulement la position relative des scores entre les méthodes peut être directement interprétée. Ainsi, le tableau 7.2 peut seulement être lu verticalement, colonne par colonne. En effet, les scores de Kendall dépendent du nombre de documents classés et du nombre de *tails*. La valeur absolue du W de Kendall ne peut pas être interprétée sans considérer la probabilité de trouver cette valeur dans les classements non corrélés. Le niveau de confiance est le complément de cette probabilité. Le tableau 7.2 seulement montre les valeurs avec un niveau de confiance d'au moins 90%. Il évalue la probabilité de la corrélation entre le classement produit par les méthodes et le classement de référence. Le tableau montre que *FLADO-tw* est la seule méthode qui a produit 14 classements sur 16 avec une probabilité $> 90\%$ d'avoir une corrélation avec le classement de référence. Les deux classements non corrélés ont été produits pour les plus longues requêtes " *nuclear factor of activated T cells* " impliquant une préposition et " *human immunodeficiency virus type 1* ". Je reviendrais sur ce phenomène plus tard. Il est très clair que *FLADO-tw* améliore *TW* sur toutes les requêtes. De cette manière nous montrons que Cortex est adapté pour résoudre les liens dans les classements de *TW*. Réciproquement, une combinaison similaire des métriques dégrade les classements de Heads, tandis que les deux méthodes *TW* et *Head* considérés séparément obtiennent des scores du W de Kendall similaires sur plusieurs requêtes où la catégorie est principalement déterminée par le mot de l'entête. Si l'on regarde les métriques de Cortex séparément, on voit que leurs résultats sont plus faibles que ceux des méthodes *Head* et *TW*. Toutefois il est intéressant d'observer que les metriques A, D et O sont nécessaires pour couvrir l'ensemble complet de requêtes où la combinaison de FLADO est significative. Il est également intéressant de noter que la méthode *Comp*, basée sur des relations sémantiques strictes, est principalement performante sur les requêtes où les métriques de Cortex n'obtiennent pas des bons scores, comme par exemple " *nuclear factor, T lymphocyte, activated T cell* ". Cela indique qu'une approche hybride est en effet souhaitable pour l'amélioration du raffinement de requêtes et les systèmes TermWatch et Cortex sont en effet complémentaires pour cette tâche. On regarde maintenant les requêtes où l'approche hybride n'a pas été aussi performante que prévu, c'est-à-dire, où les méthodes indépendantes ont obtenu de meilleurs classements. La méthode *Head* surpasse de manière significative toutes les autres méthodes sur la requête " *Epstein-Barr virus* " étant donné que le mot d'en-tête " *virus* " caractérise les termes de cette catégorie GENIA, c'est-à-dire, presque tous les termes de cette catégorie incluent le mot " *virus* ". Ainsi, compter les occurrences des mots de cette entête dans les documents est équivalent à compter les occurrences des termes de cette catégorie. Il y a cependant une différence entre le classement produit par *Head* et celui de référence car le dernier enregistre la présence simple d'un terme dans un document même si le terme a des occurrences multiples. La fonction *Tf.idf* est la seule qui a obtenu un classement significativement corrélé sur la requête " *human immunodeficiency virus type 1* "

malgré l'ambiguïté du sujet de cette requête : *1* et *virus type 1*. Le classement par TermWatch de base (W=0, 60) et Head (W=0, 51) a suivi sur cette requête. TermWatch est plus performant que Head parce qu'il utilise l'information de fonction grammaticale pour l'extraction des termes. Par conséquent, il peut identifier l'entête correcte des termes de la requête. Étonnamment, les autres relations de TermWatch (*Comp, Var*) sont également basées sur une telle extraction de termes mais utilisent une liste de raffinage des relations moins performantes que *tf.idf* sur cette requête. Ceci peut être expliqué par le fait que plus les termes sont longs, moins il y a de relations pour regrouper dans TermWatch. En effet, avec des termes de requêtes plus courtes, comme " *Jurkat Cell* " ou " *activated Cell* " où le mot d'entête est plus commun, les relations de TermWatch sont plus performantes. Une requête n'a pas été incluse au tableau (" *nuclear factor of activated T cells* ") parce qu'aucune méthode n'a atteint le niveau de confiance de 90%. Cette requête a la particularité de contenir une préposition. Il est intéressant d'observer que la méthode la plus performante sur cette requête est la variante *tf.idf* avec QE (W=0, 51), sans pour autant atteindre une probabilité convaincante d'être corrélée avec le classement de référence.

Queries :	B cell	protein kinase C	T cell	NF-kappa B	Jurkat cell	transcription factor	T lymphocyte	Epstein-Barr virus	nuclear extract	glucocorticoid receptor	human monocyte	nuclear factor	Jurkat T cell	activated T cell	human immunodeficiency virus type 1
Head	**0.61**	**0.69**	**0.58**		0.68	**0.60**		**0.86**	0.65		0.59	0.58	0.62		
FLADO-head	0.58	0.62	**0.60**		0.69	**0.58**		**0.77**	0.61				0.63		
A		0.63		**0.67**	0.65					0.59					
D	0.58	0.63	**0.57**	**0.62**	0.69										
F		0.63		**0.67**	0.65					0.59					
L		0.62		**0.67**	0.65	0.56									
O	0.56	0.63	0.55	**0.65**	0.67	0.55									
FLADO	0.57	0.61	**0.57**	**0.63**	**0.70**	**0.57**									
Comp			0.57				0.61					0.65		**0.77**	
Var			**0.60**		**0.78**							0.63	0.64		
TW	**0.74**		**0.72**	0.58	**0.77**	**0.60**	0.65	**0.75**	0.63	0.60	**0.75**		0.67	0.75	
FLADO-tw	**0.88**	0.67	**0.88**	**0.61**	**0.88**	**0.73**	**0.80**	**0.73**	**0.84**	**0.73**	0.68	**0.75**	**0.80**	**0.77**	
QE		0.65		**0.64**	**0.70**	0.54							0.61		
tf.idf								0.68					**0.70**		0.70

TAB. 7.2: *Scores du W de Kendall par requête. Seulement les scores avec un niveau de confiance* $\geq$ 90% *apparaissent. Les valeurs avec un niveau de confiance entre* 90% *et* 95% *sont en italique. Les valeurs en gras ont un niveau de confiance* > 99%*. Reproduit de (SanJuan et al., 2007), p. 261.*

7.5 Conclusion

La tâche présentée, que nous avons nommé *Semantic Query Expansion oriented Document Ranking* (SQEDR) est tout à fait nouvelle et n'a pas été traitée dans les campagnes TREC[6] (Buckley, 2005). Les résultats obtenus sur le corpus GENIA montrent

[6] http://trec.nist.gov

que de tels classements peuvent être rapprochés combinant l'extraction de termes MWT et la représentation des textes par sac-de-mots. Dans la récente évaluation TREC'05, (Liu et Yu, 2005) ont utilisé la désambiguïsation de mots et l'expansion sémantique de termes des requêtes dans la tâche de recherche documentaire. La désambiguïsation a été d'abord appliquée aux termes multi-mots de la requête afin de déterminer le sens exact des mots constituants dans le contexte de la requête. Ceci est fait en utilisant l'information disponible dans WordNet. Quand cette procédure échoue, les auteurs font appel à une recherche dans le Web pour desambigüiser. Après que la désambiguïsation est été exécutée, les termes sémantiquement associés au sens choisi (*synsets*) à partir de WordNet sont utilisés pour augmenter les termes de la requête. Ceci dit, l'expansion de la requête par cette technique est fortement dépendante de la couverture de mots de WordNet dans le corpus. Nous pensons que l'approche SQEDR pourrait être utile dans la tâche TREC standard. Nous travaillons également dans les graphes du corpus général MEDLINE. Un raffinement de requêtes peut être effectué sur ce corpus en utilisant le thesaurus MeSH[7] et l'UMLS [8]. Cependant, ces deux contiennent seulement les termes d'un vocabulaire contrôlé (termes fabriqués manuellement), qui ne sont pas forcement présents dans les résumés MEDLINE. Notre approche SQR pourrait combler le vide entre les termes réels des textes et un vocabulaire contrôlé.

[7] *Medical Subject Headings* : associé aux descripteurs MEDLINE, http://www.nlm.nih.gov/mesh
[8] Unified Medical Language System.

Chapitre 8

Retour à la physique statistique : l'énergie textuelle

M.C. Escher. Puddle. Bois en trois couleurs, 1952.
All M.C. Escher works (c) 2007 The M.C. Escher Company - the Netherlands.
All rights reserved. Used by permission. www.mcescher.com

Après avoir passé en revue toute une palette de méthodes issues de la physique statistique, des perceptrons, du modèle vectoriel de textes, des méthodes probabilistes et des algorithmes de résumé automatique, je finirai cette dissertation par un chapitre que revient à la source de mes recherches, c'est-à-dire... à la physique statistique. Si surprenante qui puisse paraître, entre le Traitement Automatique de la Langue Naturelle et la physique statistique il y a des ponts à traverser. Il suffit tout simplement de les trouver ou au défaut, de les imaginer. Dans cette dernière partie, je présenterai une approche de réseaux de neurones inspirée de la physique statistique pour étudier des problèmes fondamentaux du TAL. L'algorithme modélise un document comme un système de neurones où l'on déduit l'énergie textuelle. Nous avons appliqué cette approche aux problèmes de résumé automatique (générique ou guidé par une thématique) et à la détection de frontières thématiques. Les résultats sont très encourageants, et les perspectives assez séductrices.

Ces travaux font partie de la thèse de Silvia Fernández, financée par le Conacyt[1] (Mexique) et en collaboration avec le Laboratoire de Physique de Matériaux[2] Université Henri Poincaré, Nancy. L'ensemble de résultats de cette étude a été publié dans les mémoires des congrès TALN'07 (Fernandez et al., 2007a) à Toulouse et MICAI'07 (Fernandez et al., 2007b) à Aguascalientes (Mexique).

8.1 L'approche de Hopfield

La contribution la plus importante de Hopfield à la théorie de Réseaux de Neurones a été l'introduction de la notion d'énergie issue de l'analogie avec les systèmes magnétiques. Un système magnétique est constitué d'un ensemble de N petits aimants appelés spins. Ces spins peuvent s'orienter selon plusieurs directions. Le cas le plus simple est représenté par le modèle d'Ising qui considére seulement deux directions possibles : vers le haut (↑, +1 ou 1) ou vers le bas (↓, -1 ou 0). Le modèle d'Ising a été utilisé dans une grande variété de systèmes qui peuvent être décrits par des variables binaires (Ma, 1985). Un système de N unités binaires possède $\nu = 1, ..., 2^N$ configurations (patrons) possibles. Dans le modèle de Hopfield les spins correspondent aux neurones qui interagissent selon la règle d'apprentissage d'Hebb[3] :

$$J^{i,j} = \sum_{\mu=1}^{P} s_\mu^i s_\mu^j \tag{8.1}$$

s^i et s^j sont les états des neurones i et j. Les autocorrelations ne sont pas calculées ($i \neq j$). La sommation porte sur les P patrons a stocker. Cette règle d'interaction est locale, car $J^{i,j}$ dépend seulement des états des unités connectées. Ce modèle est connu aussi comme mémoire associative. Elle possède la capacité de stocker et de récupérer un certain nombre de configurations du système, car la règle de Hebb transforme ces configurations en attracteurs (minimaux locaux) de la fonction d'énergie (Hopfield, 1982) :

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} s^i \; J^{i,j} \; s^j \tag{8.2}$$

L'énergie est fonction de la configuration du système, c'est-à-dire, de l'état (d'activation ou non activation) des unités. Si on présente un patron ν, chaque spin subira un champ local $h^i = \sum_{j=1}^{N} J^{i,j} s^j$ induit par les autres N spins (voir figure 8.1).

Les spins s'aligneront selon h^i pour restituer le patron stocké qui est le plus proche au patron présenté ν. Hopfield a démontré que l'énergie de ce système, définie par (8.2), diminue toujours pendant le processus de recupération. Je ne devélopperai pas ici la

[1] http://www.conacyt.mx

[2] http://www.lpm.u-nancy.fr

[3] Hebb (Hertz et al., 1991) a suggéré que les connexions synaptiques changent proportionnellement à la corrélation entre les états des neurones, comme a été dit dans la Section 1.2.

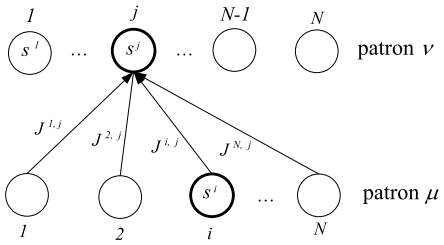


FIG. 8.1: *Champ h_i subi par le spin $s_j, \in$ la chaîne (patron) ν produit par les autres N spins $\in \mu$.*

méthode de recupération de patrons[4], car notre intérêt va porter sur la distribution et les propriétés de l'énergie du système (8.2). Cette fonction monotone et décroissante avait été utilisée uniquement pour montrer que l'apprentissage est borné. D'un autre côté, le modèle vectoriel de textes (Salton et McGill, 1983) transforme un document dans un espace adéquat où une matrice S contient l'information du texte sous forme de sacs de mots. On peut considérer S comme l'ensemble des configurations d'un système dont on peut calculer l'énergie.

8.2 L'énergie textuelle : une nouvelle mesure de similarité

Les documents sont pré-traités avec des algorithmes classiques de filtrage de mots fonctionnels[5], de normalisation et de lemmatisation (Porter, 1980; Manning et Schütze, 1999) afin de réduire la dimensionnalité. Une représentation en sac de mots produit une matrice $S_{[P\times N]}$ de fréquences/absences composée de $\mu = 1, \cdots, P$ phrases (lignes) ; $\vec{\sigma}_\mu = \{s^1_\mu, \cdots, s^i_\mu, \cdots, s^N_\mu\}$ et un vocabulaire de $i = 1, \cdots, N$ termes (colonnes).

$$S = \begin{pmatrix} s^1_1 & s^2_1 & \cdots & s^N_1 \\ s^1_2 & s^2_2 & \cdots & s^N_2 \\ \vdots & \vdots & \ddots & \vdots \\ s^1_P & s^2_P & \cdots & s^N_P \end{pmatrix} \; ; \; s^i_\mu = \begin{cases} TF^i & \text{si le terme } i \text{ existe} \\ 0 & \text{autrement} \end{cases} \tag{8.3}$$

La présence du mot i représente un spin $s^i \uparrow$ avec une magnitude donnée par sa fréquence TF^i (son absence par $\downarrow$ respectivement), et une phrase $\vec{\sigma}_\mu$ est donc une chaîne de N spins. Nous allons nous différencier de (Hopfield, 1982) sur deux points : S est une matrice entière (ses éléments prennent des valeurs fréquentielles absolues) et nous utilisons les éléments $J^{i,i}$ car cette auto-corrélation permet d'établir l'interaction du mot i parmi les P phrases, ce qui est important en TAL. Pour calculer les interactions entre les N termes du vocabulaire, on applique la règle de Hebb, qui en forme matricielle est égale à :

$$J = S^T \times S \tag{8.4}$$

[4]Cependant le lecteur intéressé peut consulter, par exemple (Hopfield, 1982; Kosko, 1988; Hertz et al., 1991).

[5]Nous avons effectué le filtrage de chiffres et l'utilisation d'anti-dictionnaires.

Chaque élément $J^{i,j} \in J_{[N\times N]}$ est équivalent au calcul de (8.1). L'énergie textuelle d'interaction (8.2) peut alors s'exprimer comme :

$$E = -\frac{1}{2} S \times J \times S^T \tag{8.5}$$

Un élément $E_{\mu,\nu} \in E_{[P\times P]}$ représente l'énergie d'interaction entre les patrons μ et ν (figure 8.1).

Je vais maintenant expliquer théoriquement la nature des liens entre phrases que la mesure d'énergie textuelle induit. Pour cela j'utiliserai quelques notions élémentaires de la théorie des graphes. L'interprétation que je ferai repose sur le fait que la matrice (8.5) peut s'écrire :

$$E = -\frac{1}{2} S \times (S^T \times S) \times S^T = -\frac{1}{2}(S \times S^T)^2 \tag{8.6}$$

Considérons les phrases comme des ensembles σ de mots. Ces ensembles constituent les sommets du graphe. On trace une arête entre deux de ces sommets σ_μ, σ_ν chaque fois qu'ils partagent au moins un mot en commun $\sigma_\mu \cap \sigma_\nu \neq \emptyset$. On obtient ainsi le graphe $I(S)$ d'intersection des phrases (voir un exemple à quatre phrases en figure 8.2). On value ces paires $\{\sigma_1, \sigma_2\}$ que l'on appelle arêtes par le nombre exact $|\sigma_1 \cap \sigma_2|$ de mots que partagent les deux sommets reliés. Enfin, on ajoute à chaque sommet σ une arête de réflexivité $\{\sigma\}$ valuée par le cardinal $|\sigma|$ de σ. Ce graphe d'intersection valué est isomorphe au graphe $G(S \times S^T)$ d'adjacence de la matrice carrée $S \times S^T$. En effet, $G(S \times S^T)$ contient P sommets. Il existe une arête entre deux sommets μ, ν si et seulement si $[S \times S^T]_{\mu,\nu} > 0$. Si c'est le cas, cette arête est valuée par $[S \times S^T]_{\mu,\nu}$, valeur qui correspond au nombre de mots en commun entre les phrases μ et ν. Chaque sommet μ est pondéré par $[S \times S^T]_{\mu,\mu}$ ce qui correspond à l'ajout d'une arête de reflexivité. Il

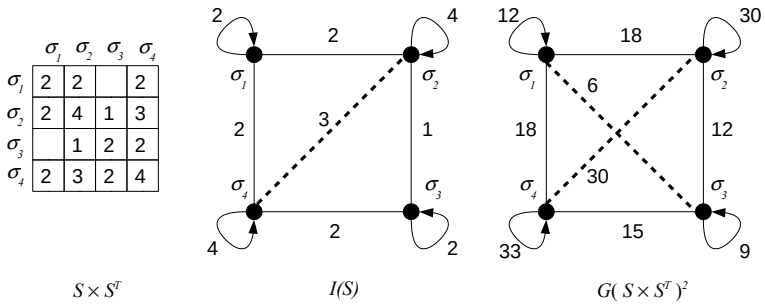


FIG. 8.2: *Graphes d'adjacence issus de la matrice d'énergie.*

en résulte que la matrice d'énergie textuelle E est la matrice d'adjacence du graphe $G(S \times S^T)^2$ dont :

- les sommets sont les mêmes que ceux du graphe d'intersection $I(S)$;
- il existe une arête entre deux sommets chaque fois qu'il existe un chemin de longeur au plus 2 dans le graphe d'intersection ;
- la valeur d'une arête :

- boucle sur un sommet σ est la somme des carrés des valeurs des arêtes adjacentes au sommet
- entre deux sommets distincts σ_μ et σ_ν adjacents est la somme des produits des valeurs des arêtes sur tout chemin de longueur 2 entre les deux sommets. Ces chemins pouvant comprendre des boucles.

De cette représentation on en déduit que la matrice d'énergie textuelle relie à la fois des phrases ayant des mots communs puisque elle englobe le graphe d'intersection, ainsi que des phrases qui partagent un même voisinage sans pour autant partager nécessairement un même vocabulaire. C'est à dire que deux phrases σ_1, σ_3 ne partageant aucun mot en commun mais pour lesquelles il existe au moins une troisième phrase σ_2 telle que $\sigma_1 \cap \sigma_2 \neq \emptyset$ et $\sigma_3 \cap \sigma_2 \neq \emptyset$ seront tout de même reliées. La force de ce lien dépend premièrement du nombre de phrases σ_2 dans leur voisinage commun, et donc du vocabulaire apparaissant dans un contexte commun.

8.3 Enertex : expériences en TAL

L'énergie textuelle peut être utilisée comme mesure de similarité dans les applications du TAL. Nous avons developpé un algorithme basé sur cette mesure de similarité appelé Enertex. De façon intuitive, cette similarité peut servir à scorer les phrases d'un document et séparer ainsi celles qui sont pertinentes de celles qui ne le sont pas. Ceci conduit immediatement à une stratégie de résumé automatique générique par extraction de phrases. Une modification en introduisant une thématique, permet de générer des résumés guidés par les besoins de l'utilisateur. Une autre approche, moins évidente, consiste à utiliser l'information de cette énergie (vue comme un spectre ou signal numérique de la phrase) et de la comparer au spectre de toutes les autres. Un test statistique peut alors indiquer si ce signal est semblable à celui d'autres phrases regroupées en segments ou pas. Ceci permet notament la détection de frontières thématiques dans un document.

8.3.1 Résumé genérique mono-document

Sous l'hypothèse que l'énergie d'une phrase μ reflète son poids dans le document, nous avons appliqué la méthode d'énergie textuelle (8.6) au résumé générique mono-document par extraction de phrases (Mani et Mayburi, 1999; Radev et al., 2002). Une modification élémentaire, permettant d'obtenir des résumés guidés par une requête ou un sujet défini par l'utilisateur[6] sera developpée dans la section suivante. L'algorithme de résumé comprend trois modules. Le premier réalise la transformation vectorielle du texte avec des processus de filtrage, de lemmatisation/*stemming* et de normalisation (c.f. chapitre 2). Le second module applique le modèle de spins et réalise le calcul de la matrice d'énergie textuelle (8.6). Nous obtenons la pondération de la phrase ν en utilisant ses valeurs absolues d'énergie, c'est-à-dire, en triant selon $\sum_\mu |E_{\mu,\nu}|$.

[6]Qui correspond au protocole des conféréences DUC (Document Understandig Conferences).

Ainsi, les phrases pertinentes seront selectionnées comme ayant la plus grande énergie absolue. Finalement, le troisième module génère les résumés par affichage et concaténation des phrases selectionnées. Le premier module repose sur le système Cortex (Torres-Moreno et al., 2002, 2000).

Nous avons opté pour l'évaluation semi-automatique avec ROUGE (Lin, 2004), qui mesure la similarité, suivant plusieurs stratégies, entre un résumé candidat (produit automatiquement) et des résumés de référence (créés par des humains). Compte tenu des difficultés de trouver un nombre suffisant de juges qui devraient générer des résumés idéaux (références), nous avons fixé un cadre expérimental de textes de petite taille et une évaluation du rappel de ROUGE-2 et SU-4. Pour les tests en français nous avons choisi les textes[7] :

– « 3-mélanges », « puces » et « J'accuse » ;

Trois textes de l'encyclopédie Wikipedia en anglais ont été analysés :

– « Lewinsky », « Québec » et « Nazca ».

Nous avons évalué les résumés produits avec ROUGE-1.5.5. Le tableau 8.1 montre les performances sur des textes en français et en anglais des systèmes MEAD, Copernic Summarizer, Enertex, Cortex et d'une *baseline* où les phrases ont été choisies au hasard[8]. Le ratio de compression a été variable (selon la taille des textes) et exprimé comme pourcentage du nombre de phrases du texte original. En gras sont affichées les performances les plus élevées, en italique celles en deuxième position (tous scores confondus). On constate que Cortex est un système résumé automatique très performant (8 premières places et 4 deuxièmes), mais Enertex n'est pas du tout mauvais (3 premières places et 6 deuxièmes). Plus encore si on réfléchit au fait que Cortex est un algorithme qui a été *pensé* depuis le début pour être un résumeur : un bon nombre de métriques assez complexes, un algorithme de décision les combinant... Ce qui n'est pas le cas d'Enertex.

Corpus	**MEAD**		**Copernic**		**Enertex**		**Cortex**		*Baseline*	
	R2	SU4	R2	SU4	R2	SU4	R2	SU4	R2	SU4
3-mélanges	⊘	⊘	0,4231	0,4348	*0,4958*	**0,5064**	**0,4967**	**0,5064**	0,3074	0,3294
Puces	⊘	⊘	**0,5775**	**0.5896**	0,5204	0,5335	*0,5356*	*0,5588*	0,3053	0,3272
J'accuse	⊘	⊘	0,2235	0,2707	*0,6146*	*0,6419*	**0,6316**	**0,6599**	0,2177	0,2615
Lewinsky	0,4756	0,4744	0,5580	0,5610	*0,5611*	*0,5789*	**0,6183**	**0,6271**	0,2767	0,2925
Québec	0,4820	*0,5169*	0,4492	0,4859	0,5095	*0,5377*	**0,5636**	**0,5872**	0,2999	0,3524
Nazca	0,4446	0,4671	0,4270	0,4495	**0,6158**	**0,6257**	*0,5894*	*0,5966*	0,2999	0,3524

TAB. *8.1:* *Rappel* ROUGE-2 *(R2) et SU4 des résumés génériques. Résumés au 25% : 3-mélanges, puces, Québec et Nazca ; résumé au 20% : Lewinsky ; résumé au 12% : J'accuse.*

Sur la graphique 8.3, on montre la moyenne des moyennes pour chaque méthode. Pour soucis de clarté, j'ai montré uniquement l'écart type correspondant à SU4 (axe vertical). On constate qu'Enertex et Cortex sont très proches en performances, et se demarquent des autres systèmes.

[7] Ces textes avaient déjà été utilisés dans le chapitre 5 sur Cortex.

[8] Faute d'espace, ni les résumés Word ni de Pertinence, tellement proches de la *baseline*, ne seront affichés. Ils sont pourtant dans la figure 8.3.

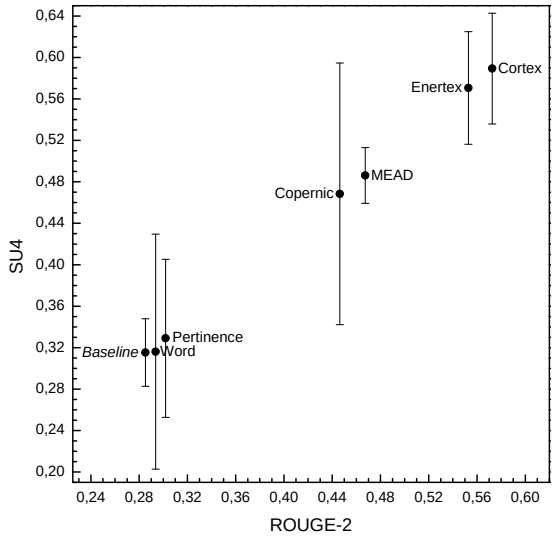


FIG. *8.3:* *Moyenne des scores moyens de rappel pour tous les textes.*

8.3.2 Résumé multi-document guidé par une thématique

La tâche principale des conférences DUC'05, '06 et '07[9], organisées par NIST, sont identiques : étant donné 45 thèmes et leurs 25 groupes de documents, il faut produire des résumés fluides de 250 mots qui répondent aux questions des thématiques. Nous avons utilisé l'énergie textuelle pour cette tâche. Nous décrivons maintenant le processus d'obtention du résumé de chaque communiqué et sa thématique. L'ensemble de 25 documents sont concatenés dans un seul long document trié chronologiquement. La thématique est ajoutée à ce document comme la dernière phrase. Un prétraitement standard (un filtrage et stemming (Porter, 1980)) lui est appliqué afin d'obtenir la matrice S (8.3). L'énergie textuelle entre la dernière phrase (la thématique) et chacune des autres phrases dans le document est calculée. Finalement, le résumé est formé avec les phrases d'une valeur absolue maximum de l'énergie.

Élimination de la redondance.

Le résumé est construit en alignant les phrases les plus pertinentes dans le document. De ce fait, dans un résumé multi-document il y a une probabilité significative de re-inclure l'information dejà présente. Pour diminuer ce problème, il faut implémenter une stratégie d'élimination de la redondance. Notre système n'inclut pas le traitement linguistique, puis notre stratégie de non-redondance consiste d'un coté, en comparer les valeurs d'énergie entre les phrases avant de les inclure, et d'un autre en utiliser l'information de la longueur des phrases.

Nous supposons que (dans de grands corpus) la probabilité que deux phrases aient les mêmes valeurs d'énergie est très faible. Alors, nous avons détecté la présence de doublons (avec exactement la même valeur d'énergie). On a observé em-

[9] Voir le chapitre 6 pour plus de détails concernant les conférences DUC.

piriquement que dans un corpus suffisamment grand, deux phrases 1 et 2, avec la même valeur d'énergie d'interaction E_1 et E_2 par rapport à la thématique sont exactement égales. Est-ce qu'on peut aller plus loin et détecter avec ce même critère, phrases égales à quelque mots près ? Pour le tester, on considère que si deux phrases partagent la plus grande partie de leurs mots, elles apportent la même information. On commence à construire le résumé avec la phrase la plus énergétique (en valeur absolue), la suivante dans le score (candidate) fera partie du résume uniquement si $|E_2 - E_1| \geq \epsilon$. E_1 est l'énergie de la phrase dite de référence. La troisième phrase candidate fera partie du résumé uniquement si $|E_3 - E_1| \geq \epsilon$ et si $|E_3 - E_2| \geq \epsilon$. Les énergies E_1 et E_2 sont considérées celles des phrases référence. En générale, une phrase candidate i sera ajoutée au résumé, si pour chaque phrase de référence $(i-1)$:

$$|E_i - E_{i-1}| = \Delta E \geq \epsilon; i = 2, 3, ... \tag{8.7}$$

Le cas contraire signifie que les énergies sont très proches avec une haute probabilité de redondance. On présente sur la figure 8.4 les valeurs du rappel du produit ROUGE-2 $\times$ SU4 pour différentes valeurs de ΔE. Le meilleur résultat sur le corpus DUC (05, 06 et 07) est obtenu avec $\Delta E \approx 0,003$. On a constaté que cela correspond aux phrases avec un ou deux mots différents.

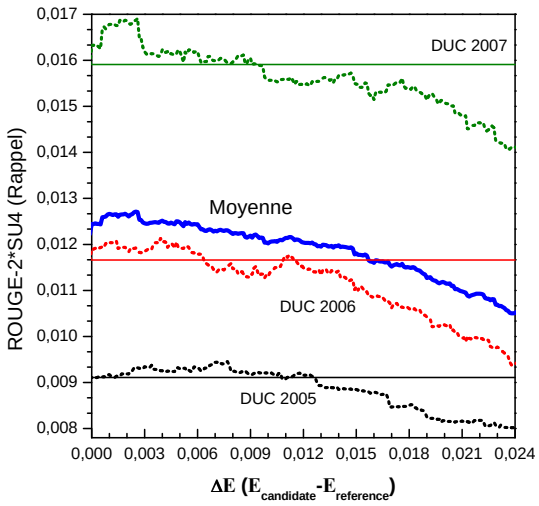


FIG. 8.4: *Elimination de la redondance : ΔE d'énergie des phrases.*

Une autre stratégie permettant de diversifier le contenu, consiste à négliger du résumé final les phrases longues (dans les groupes de documents il y a des phrases de taille comparable à celle du résumé demandé). Nous avons défini la taille maximale de phrase permise comme $k \times M$ où M = nombre moyen de mots par phrase dans les documents originaux. Nous avons fait varier k par de petits pas en mesurant à chaque moment le produit de ROUGE-2 $\times$ SU4 avec le comportement montré sur la figure 8.5. On obtient le meilleur résultat (moyen) de ROUGE autour de $k \approx 1,6$.

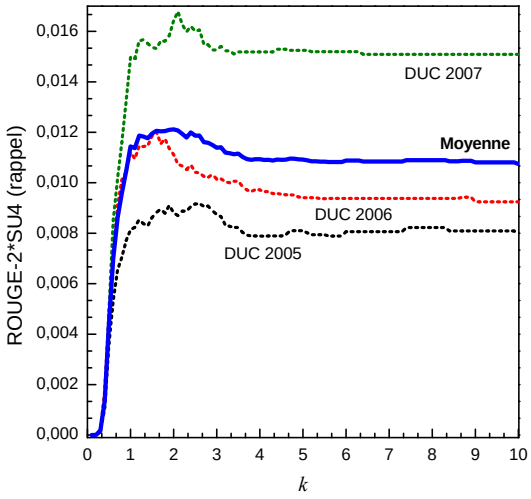


FIG. 8.5: *Elimination de la redondance : moyenne des mots.*

En pratique, nous avons fixé d'abord le paramètre $k = 1,6$ et puis le seuil d'énergie $\Delta E = 0,003$, toujours en maximisant le produit ROUGE-2×SU4.

Les figures 8.6 et 8.7 montrent la position du système Enertex (bar noire) dans l'évaluation semi-automatique ROUGE de DUC, comparé aux autres participants et la *baseline* aléatoire (avec l'identifieur NIST).

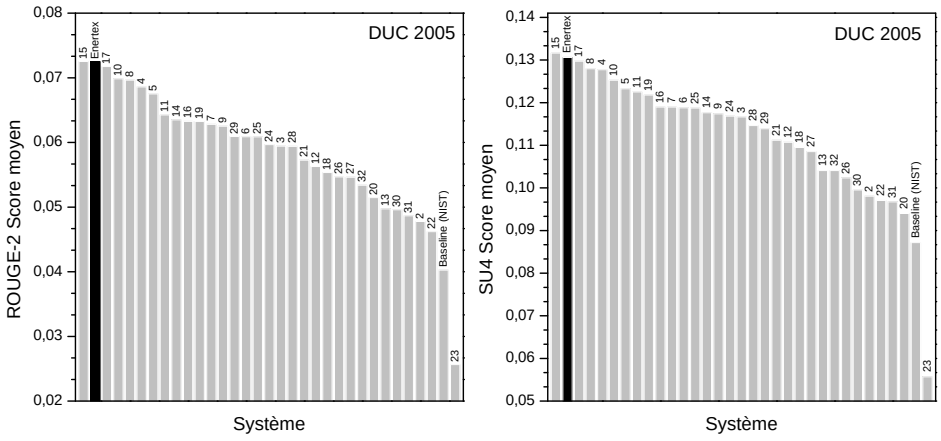


FIG. 8.6: *DUC'05 : Rappel* ROUGE-2 *et SU4 des 30 participants et la baseline.*

Dans le cas de DUC'07 (figure 8.8) le comité a inclut deux évaluations du type *baseline* (identifieurs NIST 1 et NIST 2). La première reste la même qu'en 2006 : une *baseline* genérée au hasard. La deuxième est une *baseline* dite intelligente, où un système de résumé générique a pris la place du système aléatoire. Cela explique pourquoi cette méthode a battu presque la moitié des systèmes participants.

La figure 8.9 montre la position du système Enertex (triangle) dans l'évaluation SU4 vs ROUGE-2, comparé aux autres participants de DUC'05 à '07. J'ai affiché

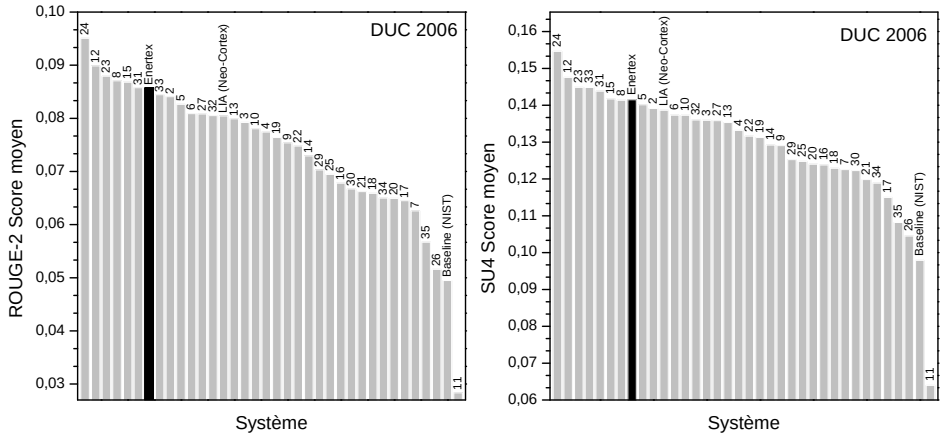


FIG. 8.7: *DUC'06 : Rappel* ROUGE-2 *et SU4 des 34 participants et la baseline.*

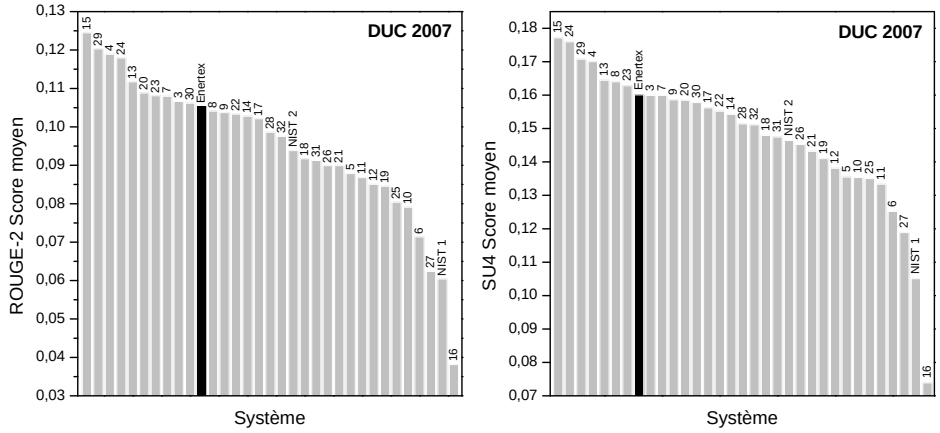


FIG. 8.8: *DUC'07 : Rappel* ROUGE-2 *et SU4 des participants et les deux baselines.*

uniquement les performances des systèmes au-dessus des *baselines*.

8.3.3 Détection de frontières thématiques

Plusieurs stratégies ont été développées pour segmenter thématiquement un texte. Elles peuvent être supervisées ou non. On trouve PLSA (Brants et al., 2002) qui estime les probabilités d'appartenance des termes à des classes sémantiques, des méthodes s'appuyant sur des modèles de Markov (Amini et al., 2000), sur une classification des termes (Caillet et al., 2004; Chuang et Chien, 2004) ou sur des chaînes lexicales (Sitbon et Bellot, 2005). De façon originale, nous avons utilisé la matrice d'énergie E (8.6). Ce choix permet de s'adapter à de nouvelles thématiques et de rester indépendant vis à vis de la langue des documents.

Nous montrons en figure 8.10 l'énergie d'interaction entre quelques phrases d'un texte composé de deux thématiques. Étant donné que (8.6) est capable de détecter et de

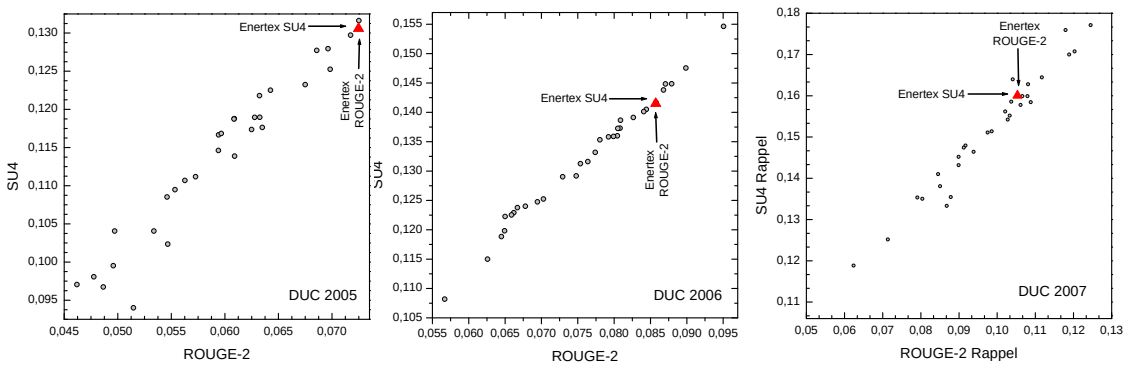


FIG. 8.9: *Aperçu du rappel SU4 vs* ROUGE-2 *des participants au-dessus des* baselines.

pondérer le voisinage d'une phrase, on peut constater une similarité entre les courbes de l'une (gras) et de l'autre thématique (pointillé).

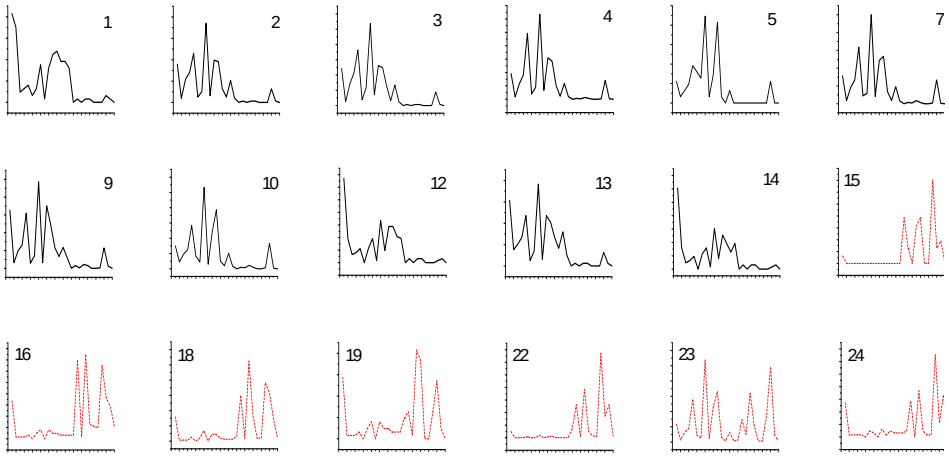


FIG. 8.10: *Énergie textuelle de « 2-mélanges ». En trait continu l'énergie des phrases de la 1ère thématique, en pointillé celle de la 2ème. Le changement d'allure des courbes entre les phrases 14-15 correspond à un changement thématique. L'axe horizontal indique le numéro de phrase dans l'ordre du document. L'axe vertical, l'énergie textuelle de la phrase affichée vs. les autres.*

Test du W de Kendall.

Pour pouvoir comparer les énergies entre elles nous introduisons le coefficient de concordance W de Kendall (Siegel et Castellan, 1988) et le calcul de sa $p-$valeur. Ils permettent de définir un test statistique de concordance entre k juges qui classent un ensemble de P objets. Nous avons utilisé ce test pour trouver les frontières thématiques entre segments. Voici le premier protocole du test que nous avons adopté.

1. Selon la nature du texte (homogène ou hétéroclite) on émet *a priori* l'une des deux hypothèses initiales H_0 qui suivent : i) la phrase $\mu + 1$ appartient à la même thématique que la phrase précédente μ ou au contraire ii) la phrase $\mu + 1$ marque une rupture avec μ.

2. On estime alors la probabilité p que l'hypothèse H_0 choisie soit vérifiée en calculant le coefficient de concordance W de Kendall entre les deux classements induits par les phrases μ et $\mu + 1$ sur les autres phrases. Le coefficient W de Kendall vaut 1 en cas d'accord total entre les classements et 0 dans la cas de désaccord total. La probabilité p est alors estimée en utilisant l'approximation de la loi du W par une loi du χ^2.
3. Finalement, si $p < 0,1$ on rejette H_0 et l'on adopte l'hypothèse alternative (son complémentaire) avec un risque p de se tromper. Il est important de préciser que la valeur seuil $0,1$ est fixée *a priori* conformément à la méthodologie statistique inférentielle.

Il s'agit donc de tests non-paramétriques qui ne requièrent aucune supposition sur une éventuelle distribution gaussienne des données. Pour chaque document, nous avons éliminé les phrases dont l'énergie est inférieure à un seuil. Ces phrases sont celles qui contiennent un nombre de mots < 2 (patrons à spins 0) ou trop longues (si l'on a suffisamment de phrases par segment), et qui induisent un fort bruit dans E. Les figures 8.12 et 8.13 montrent la détection des frontières pour les textes à 2 et 3 thématiques. Les véritables frontières sont indiquées en pointillé. Ce protocole de test, en adoptant l'hypothèse ii) comme H_0, a détecté une frontière entre les phrases 14-15 pour le texte « 2-mélanges ». Pour le texte « 3-mélanges », le test a trouvé deux frontières entre les segments 8-9 et 16-18. Dans les deux cas, cela correspond effectivement aux frontières thématiques. Une troisième (fausse) frontière a été signalée entre les phrases 23-24 du texte « 2-mélanges ». Cela mérite d'être commenté : si on regarde sur la figure 8.10 l'énergie de la phrase 23, elle est bien différente de celle des phrases 22 ou 24. La phrase 23 présente une courbe chevauchant les deux thématiques. C'est pourquoi le test ne peut pas l'identifier comme appartenant à la même classe. Le même raisonnemment tient pour toutes les fausses frontières.

Kendall en fenêtre.

Le test précedent donne des résultats accceptables mais on se demandait si l'on pouvait faire mieux. Il est clair que l'énergie d'une phrase i (et par conséquence son allure, donc sa classe) peut être faussée par les mots que la phrase i contient et qui appartient aussi à des phrases d'autres thématiques. Voir figure 8.11, par exemple. Ceci nous a amené à revoir notre protocole de test, en considérant l'information des phrases voisines à i, se trouvant à l'intérieur d'une fenêtre où le test sera effectué.

Afin de comparer les énergies entre elles-mêmes nous avons utilisé le coefficient de corrélation de Kendall τ. Etant donné deux phrases μ et ν, nous estimons la probabilité $P[\mu \neq \nu]$ d'être dans des thématiques différentes par la probabilité de $[\tau(x, y) > \tau(E_{\mu,.}, E_{\nu,.})]$. Ceci est fait en utilisant un approximation normale de la loi de Kendall τ valide si les vecteurs $E_{\mu,.}, E_{\nu,.}$ ont plus de 30 termes. Le coefficient τ ne dépend pas des valeurs exactes de l'énergie, seulement sur leur classements dans les vecteurs $E_{\mu,.}, E_{\nu,.}$. Fondamentalement, il évalue le degré de concordance entre deux classements et fait possible un test d'accord statistique non paramétrique robuste entre deux juges classifiant un ensemble de P objets en utilisant ce fait que $P[\tau(x, y) > \tau(E_{\mu,.}, E_{\nu,.})] = 1$ si les vecteurs se classant associés à $E_{\mu,.}$ et

sont 2 variables statistiquement indépendantes. Ici, les juges sont deux phrases qui classifient toutes les autres phrases basées sur l'énergie d'interaction. Nous dirons qu'il est presque sûr que deux phrases μ et ν sont dans la même thématique si $P[\mu \neq \nu] > 0,05$. Nous avons utilisé ce test pour trouver les frontières thématiques entre segments.
Comme illustré à la figure 8.11, une phrase est considérée comme la frontière d'un segment s'il est presque sûr que :

1. Elle est dans la même thématique qu'au moins 2/3 des phrases précédentes ;
2. Elle n'est pas dans la même thématique qu'au moins 2/3 des phrases suivantes.

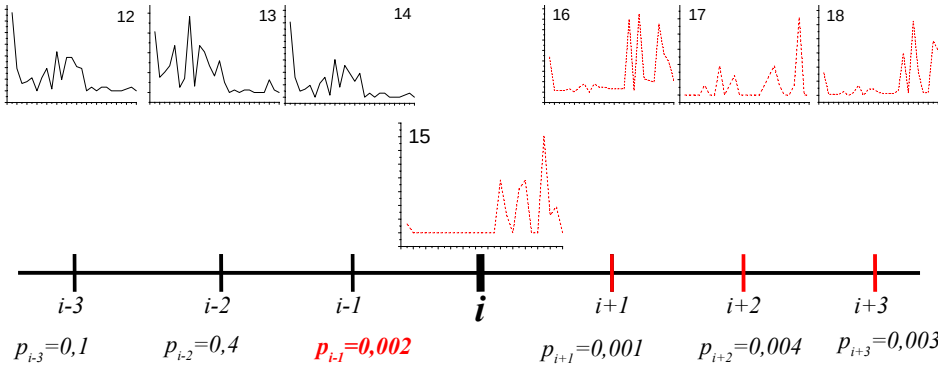


FIG. 8.11: *τ de Kendall en fenêtre. $p_{i\pm k}$ = probabilité de concordance entre $i \pm k$ et i.*

Nous avons implanté cette approche de segmentation thématique comme une fenêtre glissante de sept phrases. Pendant que la fenêtre se déplace, la phrase centrale est comparée à toutes les autres phrases dans la fenêtre (basée sur le coefficient de Kendall τ). Si une frontière est trouvée alors la fenêtre saute par-dessus les trois suivantes phrases. Nos programmes ont été optimisés avec les bibliothèques standard de Perl 5. Les figures 8.12 et 8.13 montrent la détection des frontières pour les textes avec 2 et 3 thématiques. Les vraies frontières sont indiquées en ligne pointillée. Pour le texte *3-mélanges*, le test a trouvé deux frontières entre les segments 8-9 et 16-18. Dans les deux cas, cela correspond en effet aux frontières thématiques. Une troisième frontière (fausse) a été indiquée entre les phrases 22-23 du texte *2-mélanges*. Elle mérite d'être commenté. Si nous regardons la figure 8.10 nous pouvons noter que l'énergie de la phrase 23 est très différent de cela des phrases 22 ou 24. La phrase 23 présente une courbe recouvrant les deux thématiques. C'est la raison pour laquelle le test ne peut pas l'identifier comme appartenant à la même classe. Ce raisonnement peut être prolongé à toutes autres frontières fausses.

On montre dans la figure 8.13 la détéction des frontières pour textes avec 3 et 2 thématiques. Pour le texte « physique-climat-chanel » le test du W de Kendall a détecté trois frontières entre les phrases 5-6 et 12-15, qui correspondent aux frontières effectives. Pour le texte en anglais à deux thématiques le test a trouvé une frontière entre les segments 44-45 qui correspond à la vraie frontière.

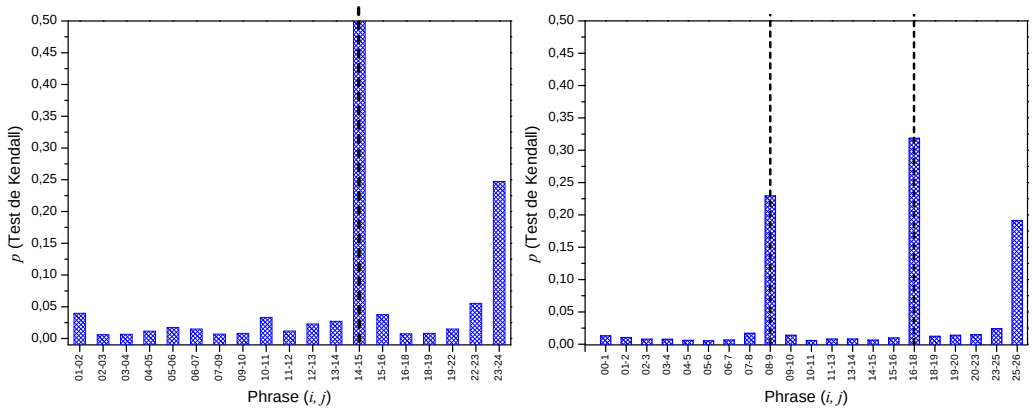


FIG. 8.12: *Détection des frontières pour le texte « 2-mélanges » (2 thématiques, à gauche) et « 3-mélanges » (3 thématiques, à droite).*

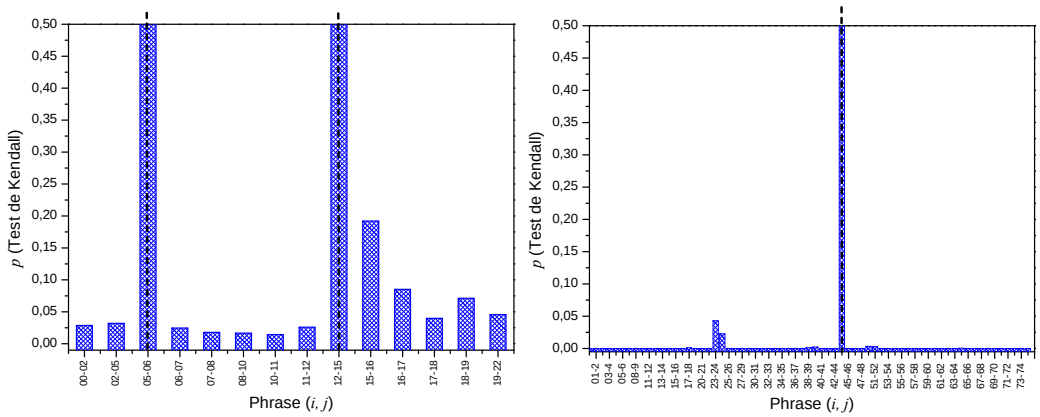


FIG. 8.13: *Détection des frontières pour le texte en français à 3 thématiques « physique-climat-chanel » à gauche et en anglais « québec-lewinsky » à droite.*

Dans une autre expérience, nous avons comparé notre système à deux autres : LCseg (Galley et al., 2003) et LIA_seg (Sitbon et Bellot, 2005), qui utilisent tous les deux des chaînes lexicales[10]. Le corpus de reférence a été construit à partir d'articles du journal LE MONDE. Il est composé d'ensembles de 100 documents où chacun correspond à la taille moyenne des segments pré-définie. Un document est composé de 10 segments extraits d'articles thématiquement différentes tirés au hasard. Compte tenu des particularités de ce corpus nous avons adopté i comme hypothèse initiale H_0. Les scores sont calculés avec WINDIFF (Pevzner et Hearst, 2002), utilisée dans la segmentation thématique. Cette fonction mesure la différence entre les frontières véritables et celles trouvées automatiquement dans une fenêtre glissante : plus la valeur est petite, plus le système est performant. LIA_seg depend d'un paramètre qui donne lieu à différentes performances (d'où la plage de valeurs affichée). Notre méthode obtient des perfor-

[10]Une chaîne lexicale relie les termes suffisamment proches dans le texte, éloignés d'une distance inférieure à une valeur fixe appelée hiatus. Classiquement, une chaîne est rompue quand elle dépasse la valeur du hiatus.

mances comparables aux systèmes dans l'état de l'art mais en utilisant bien moins de paramètres, en particulier nous ne faisons aucune supposition sur le nombre de thématiques à détecter. Le tableau 8.2 montre ces résultats et la denière colonne le nombre moyen de véritables frontières trouvées par Enertex.

Taille du segment (en phrases)	**LCseg**	**LIA_seg**	**Enertex**	(Frontières trouvées)
9-11	0.3272	(**0.3187**-0.4635)	0.4134	7.10/9
3-11	0.3837	(**0.3685**-0.5105)	0.4264	7.15/9
3-5	0.4344	(0.4204-0.5856)	**0.4140**	5.08/9

TAB. 8.2: *Mesure Windiff pour LCseg, LIA_seg et Énertex (segments de différentes tailles).*

8.4 Conclusion et perspectives

Nous avons introduit le concept d'énergie textuelle basé sur des approches des réseaux de neurones. Cela nous a permis de développer un nouvel algorithme de résumé automatique que nous avons publié (Fernandez et al., 2007a). Des tests effectués ont montré que notre algorithme est adapté à la recherche de segments pertinents. On obtient des résumés équilibrés où la plupart des thèmes sont abordés dans le condensé final. Les avantages supplémentaires consistent à ce que les résumés sont obtenus de façon indépendante de la taille du texte, des sujets abordés, d'une certaine quantité de bruit et de la langue (sauf pour la partie pré-traitement). L'algorithme d'énergie pourrait même être incorporé au système Cortex, où il jouerait le rôle d'une des métriques pilotée par un algorithme de décision. Des résumés personnalisés en fonction d'une requête de l'utilisateur ont été générés en introduisant la thématique comme la dernière phrase. Des tests sur les corpus DUC'05, '06 et '07 ont été réalisés montrant de très bonnes performances. Une autre voie intéressante est le calcul des propriétés comme la « magnétisation » d'un document. (Shukla, 1997) a étudié des phénomènes magnétiques dans les réseaux de neurones type Hopfield dont on pense se servir. On étudiera la réponse du système face à l'application d'un champ externe. Ce champ, représenté par le vecteur des termes d'un texte décrivant une thématique (thématique) sera mis en relation avec un document. Ainsi, les phrases du document pourraient ou non, s'aligner selon leur degré de pertinence par rapport à la thématique. Ceci permettrait peut être de générer des résumés avec detéction de la nouveauté, tel que définit dans la tâche pilote DUC'07. Nous avons également abordé le problème de la détection des frontières thématiques des documents. La méthode, basée sur la matrice d'énergie du système de spins, est couplée à un test statistique non-paramétrique robuste (le τ de Kendall). Les résultats sont très encourageants. Une critique de la méthode d'énergie pourrait être qu'elle nécessite la manipulation (produits, transposée) d'une matrice de taille $[P \times P]$. Cependant la représentation sous forme de graphe nous permet d'exécuter ces calculs en temps $P \log(P)$ et en espace P^2.

Chapitre 9

Bilan général et perspectives

J'ai présenté un aperçu global de mon parcours après la thèse. En partant des perceptrons et de la classification d'objets, je me suis dirigé vers le Traitement Automatique de la Langue Naturelle. La catégorisation de textes et le résumé automatique ont toujours été au cœur de mes recherches. Au cours de ce mémoire, je vous ai présenté mes différentes approches et leurs résultats. Ainsi, j'ai décidé d'étudier en profondeur les capacités de la règle d'apprentissage Minimerror et de l'algorithme Monoplan, qui grandit en apprenant. J'ai étudié théoriquement le problème de la parité N-dimensionnelle, tâche de classification très difficile pour les réseaux de neurones. Une expression pour obtenir le nombre minimum d'erreurs comme fonction du nombre des entrées a été deduite. La combinaison hybride de Minimerror et de *Fuzzy K-means* semble être prometteuse du point de vue d'un apprentissage hybride. Cette stratégie a été appliquée pour prévoir, d'une manière plutôt satisfaisante, si un site pouvait être identifié comme un dépôt de minerai ou pas. J'ai également suggéré une variation de Minimerror pour la classification non supervisée avec des séparateurs hypersphériques. Cependant il faudra encore approfondir sur cette voie.

J'ai présenté un panorama du modèle vectoriel de représentation de textes, qui permet de traiter plusieurs tâches du TAL efficacement. Deux exemples d'applications intéressantes, le routage de courriels et le traitement automatique des offres d'emploi ont ainsi été développés. Il s'agit d'applications non triviales car on y travaille avec des événements très rares (les courriels) ou avec l'information fortement non structurée (le traitement des offres d'emploi). Les méthodes de classification supervisée du type SVM mixées avec des méthodes probabilistes ont obtenu de bonnes performances. Je les ai appliquées dans d'autres tâches. La classification de textes en fonction des opinions qu'ils expriment (articles scientifiques, débats politiques, critiques de cinéma...) est une tâche cognitive très difficile même pour les personnes. La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification par rapport à une autre. Nous avons utilisé des approches de représentation numériques et probabilistes, afin de rester aussi indépendant que possible des thématiques traités. Une stratégie de fusion de plusieurs classifieurs a montré des résultats supérieurs à n'importe laquelle des méthodes individuelles utilisées. Les tâches de classification et d'identifi-

cation d'un auteur ont aussi été abordées avec des méthodes numériques. Un modèle probabiliste de cohérence interne des discours a beaucoup amélioré les résultats avec des modèles d'apprentissage préalablement développés (modélisation bayésienne, automates de Markov et processus d'adaptation). Les résultats ont été très encourageants. La fusion des hypothèses vue comme un vote de plusieurs juges pondérés par un perceptron optimal a permis de surpasser les résultats en apprentissage. Cependant, nous pensons qu'il reste encore du travail pour améliorer cette stratégie afin d'obtenir de meilleures performances en généralisation.

Une bonne partie de mes recherches a été consacrée à l'étude des systèmes de résumé automatique. Les méthodes de résumé automatique de textes ont fait un long chemin. Mais tant qu'on sera incapable de résoudre le problème de la compréhension du texte, le résumé automatique restera une approximation du résumé humain. Cependant, nos méthodes numériques (Cortex, Enertex) ont montré de bonnes performances dans des tâches de résumé mono ainsi que de résumés multi-documents guidés par une thématique. Les résultats dans les conférences DUC d'une fusion de plusieurs systèmes de résumé et d'une approche pour la détection de la nouveauté ont été au rendez-vous. Une combinaison de méthodes numériques avec une approche linguistique a été testée et cette démarche a montré des résultats intéressants : elle produit des résumés plus proches de ceux attendus par un utilisateur. La linguistique ajoute une valeur de finesse aux méthodes numériques, on obtient alors des performances améliorées. La même amélioration a été constaté lors du raffinement de réquêtes.

Finalement nous avons introduit le concept d'énergie textuelle basé sur les approches des réseaux de neurones et de la physique statistique. Le document est vu comme un système de spins où l'on peut calculer des propriétés intéressantes. Cela nous a permis de développer un nouvel algorithme de résumé automatique où les phrases du document s'alignent selon leur degré de pertinence par rapport à la thématique. L'énergie textuelle permet d'aborder également le problème de la détection des frontières thématiques des documents. Une méthode de similarité des spectres énergétiques a été developpée. La représentation sous forme de graphe nous permet d'exécuter les calculs nécessaires en temps rapides de $P\log(P)$ et en espace P^2.

J'ai également abordé d'autres domaines du TAL que, pour des raisons d'espace, je ne développerai pas dans ce mémoire mais j'en dirai quelques mots. J'ai étudié la détection de *spams* (Master Recherche de Yann Romero en 2005), la génération automatique de texte (thèse de Éric Charton qui démarre fin 2007), la compression statistique de phrases guidée par un perceptron (Master Recherche de Thierry Wazack en 2007), le traitement de petites annonces (avec des problèmatique de maximiser l'information dans un minimum de mots) et la génération et l'enrichissement automatique des patrons (thèse de Cédric Vidrequin)[1] et l'identification des entités chimiques (thèse de Florian Boudin) par des méthodes probabilistes. Finalement, un projet ANR concernant le résumé et la détection d'opinion multimédias (texte, audio, vidéo) démarre fin 2007, avec la collaboration de Georges Linares.

J'ai essayé, pendant cette période d'environ dix ans, d'être cohérent dans mes dé-

[1]Recherches financées par Semantia, `http://www.semantia.fr` et l'ANRT (contrat Cifre).

marches et d'approfondir autant que possible les méthodes d'apprentissage et de classification appliquées au traitement automatique de textes. D'après les résultats de mes recherches, j'ai constaté le fait que pour traiter le texte une analyse très fine n'est pas nécessaire. Les méthodes numériques agissent à grande échelle. Elles balaient des mots comme des billes à probabilités colorées, elles sont grossières certes, mais elles ont fait leurs preuves. Même si les mots ne sont pas des billes colorées, un de mots les plus probables (sur l'Internet) après *bille*, est *colorée*. On peut calculer cela. L'hypothèse de base de mes travaux en TAL est qu'il n'y a rien de plus concret que les textes. C'est le contenu des corpora dont on dispose avant tout. Il faut en profiter.

On ne sait pas encore écrire des programmes qui comprennent le texte. L'enjeu est difficile. La compréhension du sens d'un texte par un être humain reste encore très mal expliquée. Cependant j'ai la conviction que derrière les processus cognitifs, les motivations, l'expérience, etc. il y a une grande dose d'apprentissage et cette conviction n'est pas qu'une impression, elle est étayée par l'expérience. Un individu maîtrise la production correcte de phrases dès le jeune âge : aucune représentation formelle de la langue n'est nécessaire à ce moment là.

On peut avoir de longues discussions théoriques sur quel type de méthode est supérieur. Mais au fond, une manière rationnelle de trancher consiste à les confronter sur des données réelles. Tout au long de cette dissertation, j'ai essayé de montrer que les méthodes numériques sont performantes en utilisant une approche pragmatique : les campagnes d'évaluation nationales et internationales. Elles peuvent avoir des défauts certes, mais restent cependant un moyen assez objectif de mesurer les performances des algorithmes. Et au moins, dans les campagnes à portée de ma connaissance (DUC en résumé automatique, DEFT en catégorisation, TREC en questions-réponse), les performances des méthodes numériques surpassent (parfois de beaucoup) celles des méthodes linguistiques. Au moment de traiter de grandes masses de documents, l'analyse linguistique fine est vite dépassée par la quantité de textes à traiter. Elle perd à ce moment-là, il me semble, une bonne partie de ses atouts.

On voit des articles et des études portant sur *Jean aime Marie* et autant sur *Marie aime Jean* ou encore *Marie est aimée par Jean*. J'admets que leur analyse n'est pas simple. Le modèle vectoriel, par exemple, peut difficilement les différencier. Laissons donc la linguistique expliquer les détails et au numérique le soin de traiter une masse de 100 000 documents hétérogènes.

Ceci est la conclusion évidente qu'on pourrait tirer de façon prématurée. Mais elle n'est pas juste.

J'ai découvert tout au long de mes travaux, en particulier ceux consacrés au résumé automatique (c.f. chapitres 5 et 6) et au raffinement de requêtes (c.f. chapitre 7), qu'un système hybride combinant des approches numériques à la base et une analyse linguistique au sommet, donne de meilleures performances que les systèmes pris de façon isolée. Il produit un certain modèle explicatif de la démarche empruntée, ce qui n'est pas du tout négligeable.

Dans l'Introduction je me posais la question de savoir si la linguistique pouvait en-

core jouer un rôle dans le traitement de la langue naturelle. Je me demandais également si les méthodes numériques étaient suffisantes pour ces mêmes tâches. Je vais répondre d'abord la deuxième question et par conséquent la première car elles sont liées.

J'ai soutenu la thèse que la linguistique n'était pas un cadre approprié pour faire face à l'incertitude des langues naturelles. C'est pour cela que j'ai utilisé de façon intensive des méthodes numériques, car elles semblaient, à mes yeux, adéquates pour le TAL.

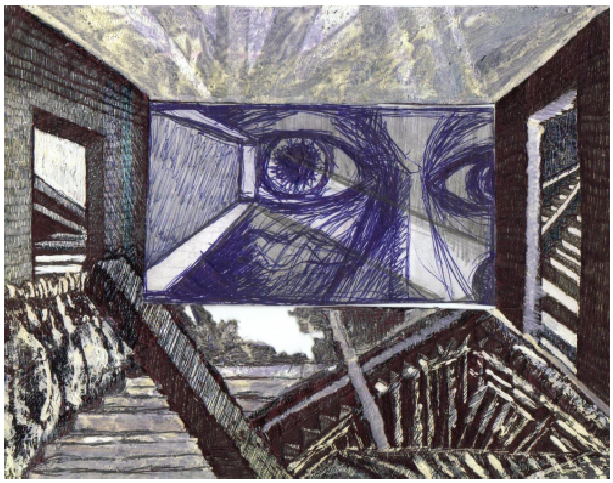
Mais cette approche a aussi ses limites.

Au-delà des promesses théoriques d'indépendance, elle est fortement dépendante des corpora annotés (souvent à la main). Les corpora sont parfois insuffisants face aux tâches complexes (c.f. campagnes DUC). Alors les unités, telles que les n-grammes, deviennent des évènements très rares. On peut, certes, pallier leur manque par des algorithmes de lissage (Good-Turing, Backoff, Katz) mais ces derniers induisent parfois des biais non évidents. Enfin, le modèle de sac de mots est une simplification exagérée qui néglige la structure de la phrase, ce qui implique une perte importante d'information.

Je reformule alors les deux questions précédentes comme ceci : *Les approches linguistiques et les méthodes numériques peuvent-elles jouer un partenariat dans les tâches du TAL ?*

La réponse est oui.

Les deux méthodes combinées peuvent combler efficacement leurs faiblesses. Elles deviennent ainsi des approches complémentaires. Cela ouvre une voie intéressante aux recherches que je compte entreprendre : la conception de systèmes TAL hybrides, notamment pour la génération automatique de texte et pour la compression de phrases. On peut difficilement envisager de dépasser le plafond auquel les méthodes numériques se heurtent sans faire appel à la finesse des approches linguistiques, mais sans négliger pour autant de les valider et de les tester sur des corpora. Car, en fin de compte, dans les tâches du TAL, seul le texte —comme celui que vous avez en face de vos yeux— est réel. Le reste est anecdotique.



Francisco Torres Moreno. Yeux, dessin, 2006.

Bibliographie

(Abdillahi et al., 2006) N. Abdillahi, P. Nocera, et J.-M. Torres-Moreno, 2006. Boîtes à outils tal pour les langues peu informatisées : le cas du somali. Actes de *JADT'06*, 697 – 705.

(Afantenos et al., 2005) S. Afantenos, V. Karkaletsis, et P. Stamatopoulos, 2005. Summarization of medical documents : A survey. *Artificial Intelligence in Medicine 2*(33), 157–177.

(Alonso et Fuentes, 2003) L. Alonso et M. Fuentes, 2003. Integrating cohesion and coherence for Automatic Summarization. Actes de *EACL'03*, 1–8. ACL, Budapest.

(Alpaydin, 1990) E. Alpaydin, 1990. *Neural Models of Incremental Supervised and Unsupervised Learning*. Thèse, Département d'Informatique, EPFL Lausanne.

(Alphonse et al., 2005) E. Alphonse, A. Amrani, J. Azé, T. Heitz, A.-D. Mezaour, et M. Roche, 2005. Préparation des données et analyse des résultats de DEFT'05. Actes de *TALN 2005/DEFT'05*, Volume 2, 95–97.

(Amati et Van Rijsbergen, 2002) G. Amati et C. Van Rijsbergen, 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems 20*(4), 357–389.

(Amini et al., 2000) M.-R. Amini, H. Zaragoza, et P. Gallinari, 2000. Learning for sequence extraction tasks. Actes de *RIAO 2000*, 476–489.

(ANSI, 1979) ANSI, 1979. *American National Standard for Writing Abstracts Z39-14*. New York, NY : ANSI, Inc.

(Aretoulaki, 1994) M. Aretoulaki, 1994. Towards a Hybrid Abstract Generation System. Actes de *Int. Conf. on New Methods in Language Processing*, 220–227. Manchester.

(Aretoulaki, 1996) M. Aretoulaki, 1996. *COSY-MATS : A Hybrid Connectionist-Symbolic Approach To The Pragmatic Analysis Of Texts For Their Automatic Smmarization*. Thèse de Doctorat, University of Manchester, Institute of Science and Technology, Manchester.

(Asterias et al., 2005) J. Asterias, E. Comelles, et A. Mayor, 2005. TXALA un analizador libre de dependencias para el castellano. *Procesamiento del Lenguaje Natural 35*, 455–456.

(Attardi, 2006) G. Attardi, 2006. Experiments with a Multilanguage Non-Projective Dependency Parser. Actes de *Tenth Conference on Natural Language Learning*. New York.

(Azé et al., 2006) J. Azé, T. Heitz, A. Mela, A.-D. Mezaour, P. Peinl, et M. Roche, 2006. Préparation de DEFT'06 (DÉfi Fouille de Textes). Actes de *SDN/DEFT'06*, Volume 2.

(Azé et Roche, 2005) J. Azé et M. Roche, 2005. Présentation de l'atelier DEFT'05. Actes de *TALN 2005/DEFT'05*, Volume 2, 99–111.

(Barzilay et Elhadad, 1997) R. Barzilay et M. Elhadad, 1997. Using lexical chains for Text Summarization. Actes de *ACL Intelligent Scalable Text Summarization*, 10–17.

(Béchet et al., 2000) F. Béchet, A. Nasr, et F. Genet, 2000. Tagging unknown proper names using decision trees. Actes de *38th Meeting ACL*, Hong-Kong, China, 77–84.

(Beauregard, 2001) S. Beauregard, 2001. *Génération de texte dans le cadre d'un système de réponse automatique à des courriels*. Thèse de Doctorat, U. de Montréal, Québec, Canada.

(Bellot et al., 2003) P. Bellot, E. Crestan, M. El-Bèze, L. Gillard, et C. D. Loupy, 2003. Coupling named entity recognition, vector-space model and knowledge bases for TREC-11, question-answering track. Actes de *TREC'02*, Volume NIST 500 251, Gaithersburg.

(Bernhard, 2007) D. Bernhard, 2007. Apprentissage non supervisé de familles morphologiques par classication ascendante hiérarchique. Actes de *TALN*, Volume 1, 367–376.

(Berthold, 1996) M. Berthold, 1996. A probabilistic extension for the DDA algorithm. *IEEE International Conference on Neural Networks 1*, 341–346.

(Berthold et al., 1995) M. Berthold, J. Diamond, et al., 1995. Boosting the performance of RBF networks with dynamic decay adjustment. Actes de *NIPS*, Volume 7, 521–528. MIT Press.

(Bezdek, 1981) J. Bezdek, 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

(Biehl et Opper, 1991) M. Biehl et M. Opper, 1991. Tilinglike learning in the parity machine. *Physical Review A 44*(10), 6888–6894.

(Bizer et al., 2005) C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, et R. Eckstein, 2005. The impact of semantic web technologies on job recruitment processes. Actes de *WI'2005, Bamberg, Germany*.

(Boudin et al., 2007) F. Boudin, B. Favre, F. Béchet, M. El-Bèze, L. Gillard, et J.-M. Torres-Moreno, 2007. The LIA-Thales summarization system at DUC-2007. Actes de *DUC'07*, Rochester, USA.

(Boudin et Torres-Moreno, 2007a) F. Boudin et J. Torres-Moreno, 2007a. NEO-CORTEX : A Performant User-Oriented Multi-Document Summarization System. Actes de *Computational Linguistics and Intelligent Text Processing*, 551–562. CICLing.

(Boudin et Torres-Moreno, 2007b) F. Boudin et J.-M. Torres-Moreno, 2007b. A Cosine Maximization-Minimization approach for User-Oriented Multi-Document Update Summarization. Actes de *RANLP*, Bulgarie, 12 pages.

(Bourse et al., 2004) M. Bourse, M. Leclère, E. Morin, et F. Trichet, 2004. Human resource management and semantic web technologies. Actes de *ICTTA'04*, 641–642.

(Brandow et al., 1995) R. Brandow, K. Mitze, et L. Rau, 1995. Automatic condensation of electronic publications by sentence selection. *Inf. Proc. and Management 31*, 675–685.

(Brants et al., 2002) T. Brants, F. Chen, et I. Tsochantaridis, 2002. Topic-based document segmentation with probabilistic latent semantic anaysis. Actes de *CIKM'02*, McLean, Virginia, USA, 211–218.

(Bruske et Sommer, 1995) J. Bruske et G. Sommer, 1995. Dynamic cell structure learns perfectly topology preserving map. *Neural Computation 7*(4), 845–865.

(Buckley, 2005) C. Buckley, 2005. Looking at limits and tradeoffs : Sabir research at trec 2005. Actes de *TREC 2005*, Gaithersburg, Maryland, USA, 13.

(Buckley et al., 1995) C. Buckley, A. Singhal, M. Mitra, et G. Salton, 1995. New retrieval approaches using SMART : TREC 4. Actes de *TREC-4*, 25–48.

(Caillet et al., 2004) M. Caillet, J.-F. Pessiot, M. Amini, et P. Gallinari, 2004. Unsupervised learning with term clustering for thematic segmentation of texts. Actes de *RIAO'04*, France, 648–657.

(Carbonell et Goldstein, 1998) J. Carbonell et J. Goldstein, 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. Actes de *21st ACM SIGIR*, 335–336. ACM Press, NY, USA.

(Carpenter et Grossberg, 1991) G. A. Carpenter et S. Grossberg, 1991. *Pattern Recognition by Self-Organizing Neural Networks*. The MIT Press.

(Carpenter et al., 1992) G. A. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, et D. B. Rosen, 1992. Fuzzy artmap : A neural network architecture for incrementalsupervised learning of analog multidimensional maps. *IEEE Trans. on Neural Networks* *3*(5), 698–713.

(Carpenter et al., 1991a) G. A. Carpenter, S. Grossberg, et J. H. Reynolds, 1991a. Artmap : Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks* *4*(5), 565–588.

(Carpenter et al., 1991b) G. A. Carpenter, S. Grossberg, et D. B. Rosen, 1991b. Art 2-a : An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks* *4*(4), 493–504.

(Carpenter et al., 1991c) G. A. Carpenter, S. Grossberg, et D. B. Rosen, 1991c. Fuzzy art : Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks* *4*(6), 759–771.

(Chakraborty et Sawada, 1996) B. Chakraborty et Y. Sawada, 1996. Fractal connection structure : effect on generalization insupervised feed-forward networks. Actes de *IEEE International Conference on Neural Networks*, Volume 1, 264–269.

(Chen et al., 2005) B. Chen, M. El-Bèze, M. Haddara, O. Kraif, et G. M. de Montcheuil, 2005. Contextes multilingues alignés pour la désambiguïsation sémantique : une étude expérimentale. Actes de *TALN'05*, Volume 1, 415–418.

(Chuang et Chien, 2004) S.-L. Chuang et L.-F. Chien, 2004. A practical web-based approach to generating Topic hierarchy for Text segments. Actes de *ACM-IKM*, Washington, 127–136.

(Cohen, 1996a) W. W. Cohen, 1996a. Learning rules that classify email. Actes de *AAAI Spring Simposium on Machine Learning in Information Access*, Minneapolis, 18–25.

(Cohen, 1996b) W. W. Cohen, 1996b. Learning to classify English text with ILP methods. Dans L. De Raedt (Ed.), *Advances in Inductive Logic Programming*, 124–143. IOS Press.

(Collobert et Bengio, 2000) R. Collobert et S. Bengio, 2000. On the convergence of svmtorch, an algorithm for large-scale regression problems. Rapport technique IDIAP-RR 00-24, IDIAP, Martigny, Switzerland.

(Collobert et Bengio, 2001) R. Collobert et S. Bengio, 2001. Support vector machines for large-scale regression problems. *Journal of Machine Learning Research 1*, 143–160.

(Collobert et al., 2002) R. Collobert, S. Bengio, et J. Mariéthoz, 2002. Torch : a modular machine learning software library. Rapport technique, IDIAP, Martigny, Switzerland.

(Cotteret et Moreau, 1969) J.-M. Cotteret et R. Moreau, 1969. *Le vocabulaire du Général de Gaulle*. Paris : Armand Colin, Presses de la Fondation nationale des sciences politiques.

(Cover, 1965) T. Cover, 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers 14*(3), 326–334.

(Cremmins, 1996) E. Cremmins, 1996. *Art of Abstracting, 2nd Edition.* Arlington, Va. : Information Resources Press.

(da Cunha et Wanner, 2005) I. da Cunha et L. Wanner, 2005. Towards the Automatic Summarization of Medical Articles in Spanish : Integration of textual, lexical, discursive and syntactic criteria. Actes de *Crossing Barriers in Text Summarization Research*. RANLP, Borovets.

(da Cunha et Wanner, 2006) I. da Cunha et L. Wanner, 2006. Resumen automático de artículos médicos en castellano : integración de técnicas de análisis textual, léxico, discursivo y sintáctico-comunicativo. Actes de *VII CLG*. Barcelona.

(da Cunha Iria et al., 2007) da Cunha Iria, S. Fernandez, P. Velázquez-Morales, J. Vivaldi, E. SanJuan, et J. M. Torres-Moreno, 2007. A new hybrid summarizer based on Vector Space model, Statistical Physics and Linguistics. Actes de *MICAI'07*, 11 pages.

(Daelemans et al., 2004) W. Daelemans, J. Zavrel, K. van der Sloot, et A. van den Bosch, 2004. Timbl : Tilburg memory based learner, version 5.1, reference guide. Rapport technique, ILK Research Group Technical Report Series.

(Damashek, 1995) M. Damashek, 1995. A gauging similarity with n-grams : Language-independent categorization of text. *Science 267*, 843–848.

(Deerwester et al., 1990) S. Deerwester, S. Dumais, G. Furnas, T. Landauer, et R. Harshman, 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science 41*(6), 391–407.

(deGruijter et McBratney, 1988) J. deGruijter et A. McBratney, 1988. A modified fuzzy k means for predictive classification. Dans H. Bock (Ed.), *Classification and Related Methods of Data Analysis*, Amsterdam, 97–104. Elsevier Science.

(Dunning, 1994) T. Dunning, 1994. Statistical identification of languages. Rapport technique MCCS 94-273, Computing Research Laboratory.

(Edmundson, 1969) H. P. Edmundson, 1969. New Methods in Automatic Extraction. *Journal of the Association for Computing Machinery 16*(2), 264–285.

(El-Bèze et al., 2005) M. El-Bèze, J.-M. Torres-Moreno, et F. Béchet, 2005. Peut-on rendre automatiquement à César ce qui lui appartient ? Application au jeu du Chirand-Mitterrac. Actes de *TALN 2005/DEFT'05*, Volume 2, 125–134.

(El-Bèze et al., 2007) M. El-Bèze, J. Torres-Moreno, et F. Béchet, 2007. Un duel probabiliste pour départager deux Présidents. *RNTI Défi fouille de textes : reconnaissance automatique des auteurs de discours - Campagne DEFT'05 (TALN'05) E10*, 148 pages.

(Fan et al., 2005) R.-E. Fan, P.-H. Chen, et C.-J. Lin, 2005. Towards a Hybrid Abstract Generation System. Actes de *Working set selection using the second order information for training SVM*, 1889–1918.

(Favre et al., 2006) B. Favre, F. Béchet, P. Bellot, F. Boudin, M. El-Bèze, L. Gillard, G. Lapalme, et J.-M. Torres-Moreno, 2006. The LIA-Thales summarization system at DUC-2006. Actes de *DUC'06 Workshop*. http ://duc.nist.gov.

(Fernandez et al., 2007a) S. Fernandez, E. SanJuan, et J.-M. Torres-Moreno, 2007a. Energie textuelle des mémoires associatives. Actes de *TALN'07*, Volume 1, 25–34.

(Fernandez et al., 2007b) S. Fernandez, E. SanJuan, et J.-M. Torres-Moreno, 2007b. Textual Energy of Associative Memories : performants applications of Enertex algorithm in text summarization and topic segmentation. Actes de *MICAI'07*, 11 pages.

(Fred et Jain, 2003) A. Fred et A. Jain, 2003. Robust data clustering. Actes de *IEEE Conference on Computer Vision and Pattern Recognition*, Volume II, 128–133.

(Freund et Schapire, 1997) Y. Freund et R. Schapire, 1997. A decision-theoretic generalization of online learning and an application to boosting. *Computer and System Sciences 55*, 119–139.

(Freund et Schapire, 1996) Y. Freund et R. E. Schapire, 1996. Experiments with a new boosting algorithm. Actes de *Thirteenth International Conference on Machine Learning*, 148–156.

(Galley et al., 2003) M. Galley, K. R. McKeown, E. Fosler-Lussier, et H. Jing, 2003. Discourse segmentation of multi-party conversation. Actes de *ACL'03*, Sapporo, Japan, 562–569.

(Gardner, 1987) E. Gardner, 1987. Maximum Storage Capacity in Neural Networks. *Europhysics Letters 4*(4), 481–485.

(Gérard et al., 2002) D. Gérard, J. Martinez, M. Samuelides, M. Gordon, F. Badran, S. Thiria, et L. Herault, 2002. *Réseaux de neurones : méthodologie et applications*. Eyrolles.

(Gillard et al., 2006) L. Gillard, L. Sitbon, E. Blaudez, P. Bellot, et M. El-Bèze, 2006. The LIA at QA@CLEF-2006. Actes de *CLEF'06 Workshop*, Alicante, 9 pages.

(Godin, Christelle, 2000) Godin, Christelle, 2000. *Contributions à l'embarquabilité et à la robustesse des réseaux de neurones en environnement radiatif. Apprentissage constructif. Neurones à impulsions*. Thèse de Doctorat, ENSAE, Toulouse, France.

(Goldstein et al., 1999) J. Goldstein, J. Carbonell, M. Kantrowitz, et V. Mittal, 1999. Summarizing text documents : sentence selection and evaluation metrics. Actes de *22nd ACM SIGIR*, 121–128. Berkeley.

(Gordon, 1996) M. Gordon, 1996. A convergence theorem for incremental learning with real-valuedinputs. *IEEE ICNN'96 1*, 381–386.

(Gordon et Berchier, 1993) M. Gordon et D. Berchier, 1993. Minimerror : A perceptron learning rule that finds the optimal weights. Actes de *ESANN*, Brussels, 105–110.

(Gordon et Grempel, 1995) M. Gordon et D. Grempel, 1995. Optimal learning with a temperature dependent algorithm. *Europhysics Letters 29*(3), 257–262.

(Gorman et al., 1988) P. Gorman et al., 1988. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks 1*(1), 75–89.

(Grefenstette, 1996) G. Grefenstette, 1996. Comparing two language identification schemes. Actes de *JADT'96*, Rome, Italia, 263–268.

(Grilheres et al., 2004) B. Grilheres, S. Brunessaux, et P. Leray, 2004. Combining Classifiers for Harmful Document Filtering. Actes de *RIAO'04*, Avignon, 173–185.

(Hajicova et al., 1995) E. Hajicova, H. Skoumalova, et P. Sgall, 1995. An Automatic Procedure for Topic-Focus Identification. *Computational Linguistics 21*(1), 81–94.

(Hasenjäger et Ritter, 1999) M. Hasenjäger et H. Ritter, 1999. Perceptron Learning Revisited : The Sonar Targets Problem. *Neural Processing Letters 10*(1), 17–24.

(Hatzivassiloglou et McKeown, 1997) V. Hatzivassiloglou et K. R. McKeown, 1997. Predicting the semantic orientation of adjectives. Actes de *8^{th} conference on European chapter of the Association for Computational Linguistics*, Morristown, NJ, USA, 174–181.

(Hertz et al., 1991) J. Hertz, A. Krogh, et G. Palmer, 1991. *Introduction to the theorie of Neural Computation*. Redwood City, CA : Addison Wesley.

(Hofmann, 1999) T. Hofmann, 1999. Probabilistic latent semantic analysis. Actes de *2nd ACM Conference on Research and Development in Information Retrieval*, 50–57.

(Hopfield, 1982) J. Hopfield, 1982. Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences of the USA 9*, 2554–2558.

(Hornik et al., 1989) K. Hornik, M. Stinchcombe, et H. White, 1989. Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 359–366.

(Hovy et al., 2005) E. Hovy, C. Lin, et L. Zhou, 2005. Evaluating DUC 2005 using Basic Elements. Actes de *DUC'05 (HLT/EMNLP)*.

(Hovy et al., 2006) E. Hovy, C. Lin, L. Zhou, et J. Fukumoto, 2006. Automated Summarization Evaluation with Basic Elements. Actes de *5^{th} Conference LREC*.

(Huot, 2000) F. Huot, 2000. Copernic summarizer ou la tentation de l'impossible. *Québec Micro 6.12*(12), 61–64.

(Ibekwe-SanJuan, 2007) F. Ibekwe-SanJuan, 2007. *Fouille de textes*. Paris, France : Hermès-Lavoisier.

(Illouz et al., 2000) G. Illouz, B. Habert, S. Fleury, H. Folch, S. Heiden, P. Lafon, et S. Prévost, 2000. Profilage de textes : cadre de travail et expérience. Actes de *JADT 2000*, Lausanne, 163–170.

(Jake D. Brutlag, 2000) C. M. Jake D. Brutlag, 2000. Challenges of the email domain for text classification. Actes de *17^{th} International Conference on Machine Learning*, San Francisco, CA, USA, 103–110.

(Jalam et Chauchat, 2002) R. Jalam et J.-H. Chauchat, 2002. Pourquoi les N-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des N-grammes caractéristiques. Actes de *JADT 2002*, St-Malo, France, 13–15.

(Joachims, 1998) T. Joachims, 1998. Text categorization with support vector machines : Learning with many relevant features. Actes de *10^{th} ECML*, Chemnitz, 137–142.

(Joachims, 1999) T. Joachims, 1999. Transductive inference for text classification using support vector machines. Actes de *16th ICML*, San Francisco, 200–209.

(Karouia et al., 1995) M. Karouia, R. Lengelle, et T. Denoeux, 1995. Performance analysis of a MLP weight initialization algorithm. Actes de *ESANN*, 347–352.

(Katz, 1987) S. M. Katz, 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing 35*, 400–401.

(Kessler et al., 2004a) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2004a. Classification thématique de courriels. Actes de *Journée ATALA sur les nouvelles formes de communication écrite*, Paris. ATALA.

(Kessler et al., 2004b) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2004b. Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé. Actes de *VSST 2004*, Volume B, Toulouse, 493–504.

(Kessler et al., 2006) R. Kessler, J. M. Torres-Moreno, et M. El-Bèze, 2006. Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage. *RSTI-ISI 11*, 93–112.

(Kessler et al., 2007) R. Kessler, J.-M. Torres-Moreno, et M. El-Bèze, 2007. E-gen : Automatic job offer processing system for human ressources. Actes de *MICAI*, 11 pages.

(Kijsirikul et Ussivakul, 2002) B. Kijsirikul et N. Ussivakul, 2002. Multiclass support vector machines using adaptive directed acyclic graph. Actes de *International Joint Conference on Neural Networks*, Volume 1, Lausanne, 980–985.

(Kim et Park, 1995) J. Kim et S. Park, 1995. The geometrical learning of binary neural networks. *Neural Networks, IEEE Transactions on 6*(1), 237–247.

(Kiritchenko et Matwin, 2001) S. Kiritchenko et S. Matwin, 2001. Email classification with co-training. Actes de *Conference of the Centre for Advanced Studies on Collaborative research*, Toronto, Ontario, Canada.

(Kohonen, 1982) T. Kohonen, 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*(1), 59–69.

(Kosko, 1988) B. Kosko, 1988. Bidirectional associative memories. *IEEE Transactions Systems Man, Cybernetics 18*, 49–60.

(Kosseim et al., 2001) L. Kosseim, S. Beauregard, et G. Lapalme, 2001. Using information extraction and natural language generation to answer E-mail. *LNCS 1959*, 152–163.

(Kosseim et Lapalme, 2001) L. Kosseim et G. Lapalme, 2001. Critères de sélection d'une approche pour le suivi automatique du courriel. Actes de *8e TALN*, 357–371.

(Kuhn et De Mori, 1995) R. Kuhn et R. De Mori, 1995. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*(5), 449–460.

(Kupiec et al., 1995) J. Kupiec, J. Pedersen, et F. Chen, 1995. A trainable document summarizer. Actes de *18th ACM SIGIR*, 68–73. ACM Press, New York.

(Labbé, 1990) D. Labbé, 1990. *Le vocabulaire de François Mitterrand*. Paris : Presses de la Fondation Nationale des Sciences Politiques.

(Land et Doig, 1960) A. Land et A. Doig, 1960. An Automatic Method of Solving Discrete Programming Problems. *Econometrica 28*, 497–520.

(Lebart, 2004) L. Lebart, 2004. Validité des visualisations de données textuelles. Actes de *JADT'04*, France, 708–715.

(Lewis et Ringuette, 1994) D. D. Lewis et M. Ringuette, 1994. A comparison of two learning algorithms for text categorization. Actes de *SDAIR-94*, Las Vegas, US, 81–93.

(Lin et Hovy, 1997) C. Lin et E. Hovy, 1997. Identifying Topics by Position. Actes de *ACL Applied Natural Language Processing Conference*, 283–290. Washington.

(Lin, 2004) C.-Y. Lin, 2004. ROUGE : A Package for Automatic Evaluation of Summaries. Dans S. S. Marie-Francine Moens (Ed.), *Text Summarization Branches Out : Proceedings of the ACL-04 Workshop*, Barcelona, Spain, 74–81.

(Liu et Yu, 2005) S. Liu et C. Yu, 2005. University of Illinois Chicago at TREC 2005. Actes de *Text Retrieval Conference (TREC 2005)*, Gaithersburg, Maryland, USA, 7 pages.

(Luhn, 1958) H. Luhn, 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development 2*(2), 159–165.

(Ma, 1985) S. Ma, 1985. *Statistical Mechanics*. Philadelphia, CA : World Scientific.

(Mani, 2001) I. Mani, 2001. *Automatic Summarization*. John Benjamins Publishing Co.

(Mani et Bloedorn, 1998) I. Mani et E. Bloedorn, 1998. Machine learning of generic and user-focused summarization. Actes de *AAAI'98/IAAI'98*, Menlo Park, 820–826.

(Mani et Mayburi, 1999) I. Mani et M. Mayburi, 1999. *Advances in automatic text summarization*. The MIT Press, U.S.A.

(Mani et Wilson, 2000) I. Mani et G. Wilson, 2000. Robust temporal processing of news. Actes de *38^{th} Association for Computational Linguistics*, Morristown, NJ, USA, 69–76.

(Mann et Thompson, 1987) W. Mann et S. Thompson, 1987. *Rhetorical Structure Theory : A Theory of Text Organization*. U. of Southern California, Information Sciences Institute.

(Mann et Thompson, 1988) W. C. Mann et S. A. Thompson, 1988. Rhetorical structure theory : Toward a functional theory of text organization. *Text 8*(3), 243–281.

(Manning et Schütze, 1999) C. D. Manning et H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts : The MIT Press.

(Marcu, 1998) D. Marcu, 1998. *The rhetorical parsing, summarization, and generation of natural language texts*. Thèse de Doctorat, Computer Science, U. of Toronto.

(Marcu, 2000) D. Marcu, 2000. *The Theory and Practice of Discourse Parsing Summarization*. Institute of Technology, Massachusetts.

(Martin et al., 1987) W. Martin, K. Churh, et R. Patil, 1987. *Preliminary analysis of a Breadth-First Parsing Algorithm : Theoretial and Experimental Results*. Springer Verlag.

(Martinez et Estève, 1992) D. Martinez et D. Estève, 1992. The Offset algorithm : building and learning method for multilayer neural networks. *Europhysics Letters 18*(2), 95–100.

(McKeown et Radev, 1995) K. McKeown et D. Radev, 1995. Generating summaries of multiple news articles. Actes de *18th ACM SIGIR*, 74–82.

(Mel'cuk, 1988) I. Mel'cuk, 1988. *Dependency Syntax : Theory and Practice*. Albany : State University Press of New York.

(Mel'cuk, 2001) I. Mel'cuk, 2001. *Communicative Organization in Natural Language. The semantic-communicative structure of sentences*. John Benjamins, Amsterdam.

(Meunier et al., 1999) J.-G. Meunier, L. Remaki, et D. Forest, 1999. Use of classifiers in computer assisted reading and analysis of text (carat). Actes de *CISST'99*, Las Vegas.

(Michel et al., 1964) H. Michel, F. Permingeat, P. Routhier, et H. Pélissonnier, 1964. Propositions concernant la définition des unités métallifères. Actes de *Commision Scientifique à la Commission de la Carte géologique du monde, 22th IGC*, New Dehli, 149–153.

(Miller et al., 2000) E. Miller, D. Shen, J. Liu, et C. Nicholas, 2000. Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System. *Journal of Digital Information 1*(5), http ://jodi.tamu.edu/Articles/v01/i05/Miller/.

(Minsky et Papert, 1969) M. Minsky et S. Papert, 1969. *Perceptrons ; an Introduction to Computational Geometry*. MIT Press.

(Morin et al., 2004) E. Morin, M. Leclère, et F. Trichet, 2004. The semantic web in e-recruitment (2004). Actes de *ESWS'2004*, Greece.

(Morris et al., 1999) A. Morris, G. Kasper, et D. Adams, 1999. The effects and limitations of automated text condensing on reading comprehension performance. Actes de *Advances in automatic text summarization*, 305–323. The MIT Press, USA.

(Namer, 2000) F. Namer, 2000. Un analyseur flexionnel du français à base de règles. *TAL 41*(2), 247–523.

(Newman et al., 2004) E. Newman, W. Doran, N. Stokes, J. Carthy, et J. Dunnion, 2004. Comparing redundancy removal techniques for multi-document summarisation. Actes de *STAIRS*, 223–228.

(Nomoto et Nitta, 1994) T. Nomoto et Y. Nitta, 1994. A Grammatico-Statistical Approach to Discourse Partitioning. Actes de *15th ICCL*, 1145–1150. Kyoto.

(Ono et al., 1994) K. Ono, K. Sumita, et S. Miike, 1994. Abstract generation based on rhetorical structure extraction. Actes de *15th ICCL*, 344–348. Kyoto.

(Paice, 1990a) C. Paice, 1990a. Constructing literature abstracts by computer : techniques and prospects. *Information Processing and Management 26*(1), 171–186.

(Paice, 1990b) C.-D. Paice, 1990b. Another stemmer. *ACM SIGIR Forum 24*(3), 56–61.

(Pardo et al., 2004) T. Pardo, M. Nunes, et M. Rino, 2004. DiZer : An Automatic Discourse Analyzer for Brazilian Portuguese. Actes de *SBIA2004*, 224–234. São Luís.

(Passonneau et al., 2005) R. Passonneau, A. Nenkova, K. McKeown, et S. Sigleman, 2005. Applying the Pyramid Method in DUC 2005. Actes de *DUC'05 (HLT/EMNLP)*.

(Perantonis et Virvilis, 1999) S. Perantonis et V. Virvilis, 1999. Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis. *Neural Processing Letters 10*(3), 243–252.

(Peretto, 1992) P. Peretto, 1992. *An Introduction to the Modeling of Neural Networks*. Cambridge University Press.

(Pevzner et Hearst, 2002) L. Pevzner et M. Hearst, 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics 28*(1), 19–36.

(Porter, 1980) M. Porter, 1980. An algorithm for suffix stripping. *Program 14*(3), 130–137.

(Quinlan, 1986) J. Quinlan, 1986. Induction of decision trees. *Machine Learning 1*(1), 81–106.

(Quinlan, 1993) J. Quinlan, 1993. *C4. 5 : Programs for Machine Learning*. Morgan Kaufmann.

(Radev et al., 2002) D. Radev, A. Winkel, et M. Topper, 2002. Multi Document Centroid-based Text Summarization. Actes de *ACL 2002*.

(Raffin et Gordon, 1995) B. Raffin et M. Gordon, 1995. Learning and generalization with Minimerror, a temperature-dependent learning algorithm. *Neural Computation 7*(6), 1206–24.

(Rafter et al., 2000a) R. Rafter, K. Bradley, et B. Smyt, 2000a. Automated Collaborative Filtering Applications for Online Recruitment Services. Actes de *LNCS*, 363–368.

(Rafter et Smyth, 2001) R. Rafter et B. Smyth, 2001. Passive Profiling from Server Logs in an Online Recruitment Environment. Actes de *IJCAI-ITWP'01)*, USA, 35–41.

(Rafter et al., 2000b) R. Rafter, B. Smyth, et K. Bradley, 2000b. Inferring Relevance Feedback from Server Logs : A Case Study in Online Recruitment.

(Ray et al., 1997) E. J. Ray, R. Seltzer, et D. S. Ray, 1997. *The AltaVista Search Revolution*. Osborne-McGraw Hill.

(Reynar et Ratnaparkhi, 1997) J. Reynar et A. Ratnaparkhi, 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. Actes de *5^{th} Conference on Applied Natural Language Processing, Washington D.C.*, 16–19.

(Rigouste et al., 2005) L. Rigouste, C. Olivier, et Y. François, 2005. Modèle de mélange multithématique pour la fouille de textes. Actes de *TALN'05/DEFT'05*, Volume 2, 193–202.

(Robertson et al., 1996) S. Robertson, S. Walker, M. Beaulieu, M. Gatford, et A. Payne, 1996. Okapi at TREC-4. Actes de *TREC-4*, 73–97.

(Sahami, 1999) M. Sahami, 1999. *Using Machine Learning to Improve Information Access.* Thèse de Doctorat, Computer Science Department, Stanford University.

(Salton, 1971) G. Salton, 1971. *The SMART Retrieval System - Experiments un Automatic Document Processing*. Englewood Cliffs.

(Salton, 1989) G. Salton, 1989. *Automatic text processing : the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Pub (Sd), Boston, MA, USA.

(Salton et McGill, 1983) G. Salton et M. McGill, 1983. *Introduction to modern information retrieval*. Computer Science Series, McGraw Hill Publishing Company.

(SanJuan et Ibekwe-SanJuan, 2006) E. SanJuan et F. Ibekwe-SanJuan, 2006. Text mining without document context. *Information Processing and Management 42*(6), 1532–1552.

(SanJuan et al., 2007) E. SanJuan, F. Ibekwe-SanJuan, J.-M. Torres-Moreno, et P. Velázquez-Morales, 2007. Combining vector space model and multi word term extraction for semantic query refinement. Actes de *NLDB'07*, Paris, France, 12 pages.

(Savoy et Abdou, 2006) J. Savoy et S. Abdou, 2006. UniNE at CLEF 2006 : Experiments with Monolingual, Bilingual, Domain-Specific and Robust Retrieval. Actes de *CLEF'06*.

(Schapire et Singer, 2000) R. Schapire et Y. Singer, 2000. BoosTexter : A boosting-based system for text categorization. *Machine Learning 39(2/3)*, 135–168.

(Schmid, 1994) H. Schmid, 1994. Probabilistic Part-of-speech Tagging Using Decision Trees. Actes de *International Conference on New Methods in Language Processing*.

(Seffah et Meunier, 1995) A. Seffah et J.-G. Meunier, 1995. ALADIN : Un atelier génie logiciel orienté objets pour l'analyse cognitive de textes. Actes de *JADT'95*, Volume 2, 105–112.

(Shukla, 1997) P. Shukla, 1997. Response of the Hopfield-Little model in an applied field. *Physical Review E 56*(2), 2265–2268.

(Siegel et Castellan, 1988) S. Siegel et N. Castellan, 1988. *Nonparametric statistics for the behavioral sciences.* McGraw Hill.

(Silber et McCoy, 2000) H. G. Silber et K. F. McCoy, 2000. Efficient text summarization using lexical chains. Actes de *Intelligent User Interfaces*, 252–255.

(Sima'an, 2003) K. Sima'an, 2003. Empirical validity and technological viability : Probabilistic models of natural language processing. Actes de *Linguistic Corpora and Logic Based Grammar Formalisms, CoLogNET Area 6*. R. Bernardi and M. Moortgat.

(Sitbon et Bellot, 2005) L. Sitbon et P. Bellot, 2005. Segmentation thématique par chaînes lexicales pondérées. Actes de *TALN 2005*, Volume 1, 505–510.

(Soboro, 2004) I. Soboro, 2004. Overview of the TREC 2004 Novelty Track. Dans E. M. Voorhees et L. P. Buckland (Eds.), *TREC'04*, USA. NIST Special Publication : SP.

(Spriet et al., 1996) T. Spriet, F. Béchet, M. El-Bèze, C. D. Loupy, et L. Khouri, 22-24 mai, 1996. Traitement automatique des mots inconnus. Actes de *TALN 96*, Marseille, France, 170–179.

(Spriet et El-Bèze, 1998) T. Spriet et M. El-Bèze, 1998. Introduction of Rules into a Stochastic Approach for Language Modelling. *Computational Models of Speech Pattern Processing 169*, 350–355.

(Stairmand, 1996) M. Stairmand, 1996. *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval*. Thèse de Doctorat, Department of Language Engineering, UMIST Computational Linguistics Laboratory.

(Swan et Allan, 2000) R. Swan et J. Allan, 2000. Automatic generation of overview timelines. Actes de *23rd ACM SIGIR conference*, 49–56. ACM Press New York, NY, USA.

(Teufel et Moens, 2002) S. Teufel et M. Moens, 2002. Summarizing Scientific Articles - Experiments with Relevance and Rhetorical Status. *Computational Linguistics 28*(4), 409–445.

(Torres Moreno et Gordon, 1998a) J. Torres Moreno et M. Gordon, 1998a. Efficient adaptive learning for classification tasks with binary units. *Neural Computation 10*(4), 1007–1030.

(Torres-Moreno et al., 2002) J.-M. Torres-Moreno, J. Aguilar, et M. Gordon, 2002. Finding the number minimum of errors in N-dimensional parity problem with a linear perceptron. *Neural Processing Letters 1*, 201–210.

(Torres-Moreno et al., 2003) J. M. Torres-Moreno, L. Bougrain, et F. Alexandre, 2003. Database classification by hybrid method combining supervised and unsupervised learnings. Actes de *ICANN/ICONIP 2003*, 37–40.

(Torres-Moreno et al., 2007) J.-M. Torres-Moreno, M. El-Bèze, F. Béchet, et N. Camelin, 2007. Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? application au défi deft 2007. Actes de *AFIA/DEFT'07*, 119–133.

(Torres Moreno et Gordon, 1995) J.-M. Torres Moreno et M. Gordon, 1995. An evolutive architecture coupled with optimal perceptron learning. Dans M. Verleysen (Ed.), *ESANN*, Brussels, 365–370.

(Torres Moreno et Gordon, 1998b) J. M. Torres Moreno et M. B. Gordon, 1998b. Characterization of the sonar signals benchmark. *Neural Processing Letters 7*(1), 1–4.

(Torres-Moreno et al., 2000) J. M. Torres-Moreno, P. Velázquez-Morales, et J.-G. Meunier, 2000. Classphères : un réseau incrémental pour l'apprentissage non supervisé appliqué à la classification de textes. Actes de *JADT 2000*, Volume 2, Lausanne, Suisse, 365–372.

(Torres-Moreno et al., 2001) J. M. Torres-Moreno, P. Velázquez-Morales, et J.-G. Meunier, 2001. Cortex : un algorithme pour la condensation automatique des textes. Actes de *ARCo'01*, Volume 2, 365–366.

(Torres-Moreno et al., 2002) J. M. Torres-Moreno, P. Velázquez-Morales, et J.-G. Meunier, 2002. Condensés de textes par des méthodes numériques. Actes de *JADT*, Volume 2, St Malo, France, 723–734.

(Torres Moreno, Juan-Manuel, 1998) Torres Moreno, Juan-Manuel, 1998. *Apprentissage et généralisation par des réseaux de neurones : étude de nouveaux algorithmes constructifs*. Thèse de Doctorat, INPG, Grenoble, France.

(Van Rijsbergen, 1979) C. Van Rijsbergen, 1979. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA.

(Vapnik, 1982) V. N. Vapnik, 1982. *Estimation of Dependences Based on Empirical Data*. New York, USA : Springer-Verlag Inc.

(Vapnik, 1995) V. N. Vapnik, 1995. *The Nature of Statistical Learning Theory*. New York, USA : Springer-Verlag Inc.

(Vinot et al., 2003) R. Vinot, N. Grabar, et M. Valette, 2003. Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'internet. Actes de *TALN*, France, 275–284.

(Viterbi, 1967) A. J. Viterbi, 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Proc. 2*(13), 260–269.

(Vivaldi, 2001) J. Vivaldi, 2001. *Extracción de candidatos a término mediante combinación de estrategias heterogéneas*. Thèse de Doctorat, UPC, Barcelona.

(Vivaldi et Rodríguez, 2002) J. Vivaldi et H. Rodríguez, 2002. Medical term extraction using the EWN ontology. Actes de *Terminology and Knowledge Eng.*, 137–142. Nancy.

(Vossen, 1998) P. Vossen, 1998. *EuroWordNet : A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic publishers.

(Wilson et al., 2005) T. Wilson, J. Wiebe, et P. Hoffmann, 2005. Recognizing contextual polarity in phrase-level sentiment analysis. Actes de *EMNLP*, Vancouver, 347–354.

(Yousfi-Monod et Prince, 2006) M. Yousfi-Monod et V. Prince, 2006. Compression de phrases par élagage de l'arbre morpho-syntaxique. *Technique et Science Informatiques* *25*(4), 437–468.