Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage

Rémy Kessler[*] – Juan Manuel Torres-Moreno[**] – Marc El-Bèze[*]

[*] *Laboratoire d'Informatique d'Avignon / Université d'Avignon*
BP 1228, 84911 Avignon cedex 9 France
[**] *École Polytechnique de Montréal - Département de Génie informatique*
CP 6079, succ. Centre-ville Montréal (Québec) H3C 3A7 Canada
{remy.kessler, juan-manuel.torres, marc.elbeze}@univ-avignon.fr

RÉSUMÉ. *Les nouvelles formes de communication écrite (courriels, forums, chats, SMS, etc.) ont introduit des défis considérables pour leur traitement automatique. Ces données présentent des phénomènes linguistiques bien particuliers : messages trop courts, très bruités... Nous présentons des recherches destinées à créer des outils et des ressources génériques pour la classification de courriels. Nous nous attachons à traiter dans cette étude des problèmes posés par le routage précis de courriels. Après un processus de filtrage et de lemmatisation, nous utilisons la représentation vectorielle de textes avant d'effectuer la classification par des approches supervisées, semi-supervisées et non supervisées. Lors des tests, nous avons obtenu de très bonnes performances sur des corpus réalistes.*

ABSTRACT. *New forms of written communication (electronic mail, forum, chat, SMS, etc.) are new challenges for Natural Language Processing methods. These data present very particular linguistic phenomena: too short and very noised messages... This paper focuses on the development of generic tools and resources for e-mails classification. This study deals with the problems of the precise routing of e-mails. After a filtering and lemmatization step, vectorial representation of texts is used for classification purpose by means of supervised, semi-supervised and unsupervised learning techniques. Very good results are presented on realistic corpora.*

MOTS-CLÉS : *apprentissage supervisé et non supervisé, machines à vecteurs de support (SVM), fuzzy k-means, classification de textes, routage automatique de courriels.*

KEYWORDS: *supervised and unsupervised learning, support vector machines, fuzzy k-means, text classification, automatic e-mail routing.*

RSTI - ISI – 11/2006. SI stratégique, pages 93 à 112

1. Introduction

Les nouvelles formes de communication écrite posent des défis considérables aux systèmes de traitement automatique de la langue. Il est possible d'observer des phénomènes linguistiques bien particuliers comme les émoticônes[1], 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.). Dans cet article, nous introduisons 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[2].

Le traitement automatique des courriels est extrêmement difficile à cause de son caractère imprévisible (Beauregard, 2001; Kosseim *et al.*, 2001b; Kosseim *et al.*, 2001a; Cohen, 1996b) : des textes trop courts (moyenne de 11 mots par courriel dans nos corpus), régis par une syntaxe mal orthographiée et/ou pauvre. Ceci impose donc d'utiliser des outils de traitement automatique robustes et flexibles. Les méthodes d'apprentissage automatique à partir de textes (fouille de documents), permettent d'apporter des solutions partielles aux tâches évoquées. Elles semblent bien adaptées aux applications de filtrage, de routage, de recherche d'information, de classification thématique et de structuration non supervisée de corpus. Ces méthodes présentent, de surcroît, l'intérêt de fournir des réponses adaptées à des situations où les corpus sont en constante évolution ou bien contiennent de l'information dans des langues étrangères. L'objectif de cette recherche consiste à proposer l'application des méthodes d'apprentissage afin d'effectuer la classification automatique de courriels visant leur routage, en combinant techniques probabilistes et machines à vecteurs de support (*Support Vector Machines* SVM) (Vapnik, 1995; Collobert *et al.*, 2000). Des approches (Jalam *et al.*, 2002; Miller *et al.*, 2000) fondées sur des mots et des n-grammes de lettres ont été testées. La catégorisation thématique est au cœur de nombreuses applications de traitement de la langue. Ce contexte fait émerger un certain nombre de questions théoriques nouvelles, en particulier en relation avec la problématique du traitement d'informations textuelles incomplètes et/ou très bruitées (Kosseim *et al.*, 2001a).

2. Position du problème

Nous nous plaçons dans le cas où une boîte aux lettres 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

1. Symboles utilisés dans les messages pour exprimer les émotions, exemple le sourire **:-)** ou la tristesse **:-(**
2. Par exemple **kdo** à la place de cadeau, **10ko** pour dictionnaire, **A+** à plus tard, **@2m1** à demain, etc.

le service technique, ceux pour le service après vente seront redirigés vers ce dernier, etc.). Il s'agit donc de développer un système pour automatiser cette tâche.

Il existe des approches pour effectuer le traitement automatique de courriers electroniques écrits en anglais (Kiritchenko *et al.*, 2001; Kosseim *et al.*, 2001a; Cohen, 1996a; Jake D. Brutlag, 2000). Cependant, il s'est avéré difficile de trouver des travaux sur les courriels ou des corpus en français (des corpus en anglais existent cependant pour de la classification de *spams*). Pour générer les corpus d'apprentissage et de test, nous avons donc décidé de créer une adresse électronique et de l'abonner à diverses listes de diffusion[3] ou newsletters[4] de thèmes variés. Il faut noter que l'évaluation de notre système s'effectue en fonction de la liste de diffusion émettrice. De ce fait, nous nous trouvons face à un partitionnement de référence où le message ne peut appartenir à plusieurs thématiques. Les corpus réalistes qui ont été ainsi collectés (libres de *spams*) présentent un certain nombre de caractéristiques particulières qui sont reportées dans le tableau 1.

Nb. total de courriels	P = 2 000	
Nb. total de mots bruts	956 757	
Nb. de courriels avec pièce jointe	140	7 %
Nb. de courriels courts ($\leq$ 11 mots)	1 379	69 %
Nb. de courriels longs (> 11 mots)	621	31 %
Taille moyenne d'un courriel en mots		11
Nb. d'auteurs différents	298	
Auteurs ayant émis moins de 2 courriels	125	42 %
Auteurs ayant émis entre 2 et 5 courriels	83	28 %
Auteurs ayant émis entre 6 et 10 courriels	33	11 %
Auteurs ayant émis plus de 10 courriels	57	19 %

Tableau 1. *Statistiques du corpus. On remarque que la majorité des courriels sont des messages courts ($\approx$ 70 %) et que la répartition du nombre d'auteurs est assez importante (70 % des auteurs ont envoyé < 6 courriels)*

3. Méthodes

Les méthodes que nous avons retenues reposent sur la représentation vectorielle de textes, qui, même si elle est très différente d'une analyse structurale linguistique, s'avère performante et rapide (Manning *et al.*, 2000). Ces méthodes ont, par ailleurs, la propriété d'être assez indépendantes de la langue[5]. Dans le modèle vectoriel de re-

3. Football, jeux de rôles, ornithologie, cinéma, jeux vidéo, poème, humour, etc.
4. Sécurité informatique, journaux, matériel informatique, etc.
5. Nous avons appliqués nos outils sur du français mais ceux-ci pourraient être utilisés sur d'autres langues.

présentation, les textes (dans notre cas, les courriels) sont traités comme des sacs de termes. Les termes peuvent être assimilés à des mots ou à des n-grammes. 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 : ce sont les termes. D'après (Jalam *et al.*, 2002), si l'on utilise des n-grammes, on peut se passer de certains processus de normalisation. Suivent alors divers calculs élémentaires se basant sur la fréquence des termes ou des n-grammes de lettres, produisant le poids de chaque mot plein dans le courriel. Cependant, les données résultantes seront très volumineuses, clairsemées et très bruitées. Ce dernier terme, emprunté à la théorie du signal, signifie que l'information pertinente ne constitue qu'une faible partie de l'ensemble total des données disponibles (Lebart, 2004). Dans la représentation vectorielle, une matrice de segments-terme où chaque case représente la fréquence d'apparition d'un terme dans un segment, sera donc très volumineuse[6]. C'est pourquoi, des mécanismes de réduction s'avèrent indispensables. La première étape consiste à nettoyer le corpus afin de séparer l'en-tête, le corps et la pièce jointe du courrier électronique. Bien que l'en-tête contienne des méta-informations qui pourraient être utilisées pour la tâche de classification, de cette partie, nous utilisons uniquement l'information contenue dans le sujet. Ensuite, a lieu un prétraitement facultatif où des processus puissants de filtrage et de lemmatisation peuvent être déclenchés afin de réduire la dimensionnalité des matrices d'apprentissage et de test (Torres-Moreno *et al.*, 2000; Torres-Moreno *et al.*, 2001; Torres-Moreno *et al.*, 2002). La lemmatisation (même si elle est plus coûteuse en temps) et non une simple radicalisation[7] s'avère plus adaptée pour le français (Namer, 2000), qui est une langue latine à fort taux de flexion. Au cours de cette phase, en vue de réduire cette dimensionnalité, nous effectuons différents traitements que nous détaillerons par la suite. Nous obtenons donc à la fin de ce prétraitement une matrice qui sera divisée aléatoirement en sous-ensembles d'apprentissage et de test (ou généralisation), puis traitée par des méthodes d'apprentissage. Nous avons décidé d'utiliser l'algorithme *fuzzy k-means* (Bezdek, 1981; deGruijter *et al.*, 1988) pour l'apprentissage non supervisé et SVM (Joachims, 1998) pour l'apprentissage supervisé. La figure 1 illustre l'ensemble des opérations du prétraitement.

3.1. *Prétraitement*

La première partie du prétraitement a consisté à identifier dans le corpus chaque courriel différent, et à séparer le corps du message des pièces jointes. Nous travaillons pour l'instant, uniquement avec le corps du message, les pièces jointes contenant des graphiques, sons, etc. dans une grande diversité de formats, posent un problème complexe. Cette première étape génère un fichier présenté, au format XML afin d'en faciliter le parcours. Une fois cette tâche réalisée, un second processus se charge d'effectuer

6. Voir la figure 1 et l'équation 1.
7. Qui est en général bien adapté à l'anglais, par exemple.

les traitements de filtrage. Ainsi nous supprimons automatiquement la micropublicité (microspams) qui n'apporte aucune information permettant de catégoriser le courriel mais, au contraire, ajoute du bruit risquant de gêner cette 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 comme le montre l'exemple suivant :

```
___[ Pub ]_____________________________________________________
Inscrivez-vous gratuitement sur Siteweb, Le site de rencontres !
http ://www.site.com/index.php?origine=4
```

Grâce à l'existence du trait continu autour de `[ Pub ]` et de certains mots-clés, il est aisé de les enlever automatiquement sans risque de perte d'informations au niveau du corps du message. Nous avons par la suite supprimé la micropublicité propre au corpus, celle-ci se présentant, la plupart du temps, sous la forme de liens HTML vers des pages internet Yahoo. À l'aide d'un dictionnaire constitué à partir de sites[8] et décrivant les divers termes de phonécriture, nous remplaçons ceux-ci par leurs équivalents en langue française. Cette étape de "traduction" est réalisée avant la suppression de la ponctuation car beaucoup de termes phonécrits sont composés à l'aide de ponctuation.

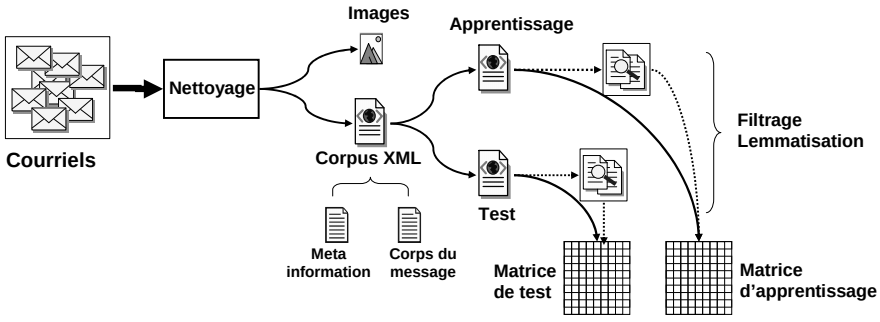

Figure 1. *Schéma général du prétraitement. Le trait pointillé du filtrage/lemmatisation indique sa nature facultative*

3.1.1. *Filtrage*

Nous effectuons dans un premier temps une suppression des verbes et des mots fonctionnels (être, avoir, pouvoir, falloir...), des expressions courantes (par exemple,

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

c'est-à-dire, chacun de...), de nombres (écrits en chiffres ou en lettres) et des symboles comme $, #, *, etc. venant bruiter la classification. Une stop-liste a été créée à la main pour cette tâche. Par la suite, nous identifions les mots composés qui représentent souvent un concept bien spécifique. Tous ces mots peuvent être répétés ou non. Ces traitements permettent de réduire la taille de la matrice et donc la complexité de son exploitation.

3.1.2. *Lemmatisation*

Ce traitement peut entraîner une réduction importante du lexique. Nous avons cependant aussi étudié l'effet de ne pas lemmatiser les corpus. La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier[9] avant de leur associer un nombre d'occurrences. Ce processus permet d'amoindrir la malédiction dimensionnelle[10] qui pose de très sérieux problèmes de représentation dans le cas des grandes dimensions. La lemmatisation permet donc de diminuer le nombre de termes qui définiront les dimensions de l'espace de représentation de termes ou espace vectoriel. D'autres mécanismes de réduction du lexique sont aussi déclenchés. Les mots composés sont repérés automatiquement à l'aide d'un dictionnaire, puis transformés en un terme unique lemmatisé en utilisant des tableaux associatifs.[11]

3.1.3. *Dictionnaire avec et sans accents*

Dans un premier temps, nous avons effectué l'ensemble des filtrages avec un dictionnaire avec accents. L'usage des caractères accentués n'étant pas systématique dans les courriels, nous avons adapté en conséquence notre dictionnaire. Nous avons amélioré la qualité de notre filtrage en utilisant un dictionnaire composé des deux précédents (avec et sans accent) : sur un corpus de 2 000 messages, $\approx$ 43 000 lemmatisations contre $\approx$ 40 000 lemmatisations précédemment. L'ensemble de ces opérations de prétraitement ayant réduit la taille du corpus d'environ 70 % (sur un corpus d'environ 950 000 occurences au départ, nous obtenons un corpus final de $\approx$ 250 000 occurences après ces opérations) et les performances ont amélioré.

4. Catégorisation

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 al.*, 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 ou non supervisé.

9. Ainsi les mots *chante, chantaient, chanté, chanteront* et éventuellement *chanteur* seront ramenés à la même forme.
10. *The curse of dimensionality.*
11. *pomme de terre* et *pommes de terre* deviennent ainsi **pomme_de_terre.**

Dans le modèle vectoriel, chaque vecteur Γ^μ appartient à une classe $g(\Gamma^\mu)$, g étant une fonction inconnue. L'ensemble d'exemples dont on dispose, appelé l'ensemble d'apprentissage, consiste en P exemples. Nous dénoterons cet ensemble Λ. Un ensemble d'apprentissage etiqueté contient P couples d'entrée-sortie $(\overrightarrow{\Gamma^\mu}, \tau^\mu); \mu = 1, \ldots, P$. Si l'ensemble n'est pas étiqueté, on disposera uniquement des P entrées $\overrightarrow{\Gamma^\mu}$ et d'aucune information concernant leur classe. Dans une approche supervisée, le classifieur a besoin de connaître les classes τ^μ afin d'approcher la fonction g. Dans une approche non supervisée, τ^μ peut être ignoré, les objets seront regroupés uniquement en fonction de leurs caractéristiques $\overrightarrow{\Gamma^\mu}$. Une autre approche intermédiaire, dite semi-supervisée, nécessite de connaître uniquement un faible nombre d'exemples de l'ensemble d'apprentissage Λ avec leur classe, afin d'effectuer des regroupements sur le reste des données.

Le prétraitement transforme le corpus en un ensemble de P vecteurs (nombre de courriels) à N dimensions (taille du lexique). Nous obtenons donc une matrice fréquentielle $\Gamma = (\overrightarrow{\Gamma^\mu}); \mu = 1, \ldots, P$ où chaque composante $(\Gamma_1^\mu, \Gamma_2^\mu, \cdots, \Gamma_N^\mu)$ contient la fréquence Γ_i^μ du terme i dans un courriel μ :

$$\mathbf{\Gamma} = \begin{bmatrix} \Gamma_1^1 & \Gamma_2^1 & \Gamma_3^1 & \ldots & \Gamma_i^1 & \ldots & \Gamma_N^1 \\ \Gamma_1^2 & \Gamma_2^2 & \Gamma_3^2 & \ldots & \Gamma_i^2 & \ldots & \Gamma_N^2 \\ \Gamma_1^3 & \Gamma_2^3 & \Gamma_3^3 & \ldots & \Gamma_i^3 & \ldots & \Gamma_N^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Gamma_1^\mu & \Gamma_2^\mu & \Gamma_3^\mu & \ldots & \Gamma_i^\mu & \ldots & \Gamma_N^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Gamma_1^P & \Gamma_2^P & \Gamma_3^P & \ldots & \Gamma_i^P & \ldots & \Gamma_N^P \end{bmatrix}, \quad \Gamma_i^\mu \in \{0, 1, 2, \ldots\} \qquad [1]$$

Un filtrage adéquat permet de ne garder que le lexique appartenant à chaque classe. Une répartition des termes en fonction de chaque classe a permis de mettre en évidence, de façon expérimentale, ce phénomène (Kessler *et al.*, 2004b).

Mesure d'erreur

Soit Λ un ensemble de P exemples avec leur classe d'appartenance τ^μ. Soit γ_1 un sous-ensemble d'apprentissage de P_1 exemples tirés au hasard, et γ_2 le sous-ensemble de test indépendant de P_2 exemples, tel que $\Lambda = \gamma_1 \cup \gamma_2$ et $P = P_1 + P_2$. Un exemple mal classé par un classifieur est un exemple dont la classe attribuée τ^μ n'est pas correcte. Ce type d'erreur peut être estimé en phase d'apprentissage ϵ_a ou de généralisation ϵ_g. Ce sont des valeurs comprises entre 0 et 1, ou exprimées en pourcentage. Pour mesurer la performance des algorithmes de classification, nous avons utilisé les expressions suivantes afin d'estimer ϵ_a et ϵ_g :

$$\epsilon_a = \frac{\text{Nb. exemples } \in \gamma_1 \text{ mal classés}}{card\{\gamma_1\}}; \epsilon_g = \frac{\text{Nb. exemples } \in \gamma_2 \text{ mal classés}}{card\{\gamma_2\}} \quad [2]$$

où $card\{\bullet\}$ représente le nombre d'éléments d'un ensemble. Ces mesures seront utilisées dorénavant dans nos tests.

5. Apprentissage non supervisé

L'algorithme Fuzzy *k-means* (Bezdek, 1981; deGruijter *et al.*, 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). *Fuzzy k-means* minimise la somme des erreurs quadratiques avec les conditions suivantes :

$$\sum_{k=1}^{c} m_{\mu k} = 1; \quad \mu = 1, \cdots, P \quad [3]$$

$$\sum_{k=1}^{c} m_{\mu k} > 0; \quad m_{\mu k} \in [0,1]; \quad k = 1, \cdots, c; \quad \mu = 1, \cdots, P \quad [4]$$

On définit alors la fonction objectif :

$$J = \sum_{\mu=1}^{P} \sum_{k=1}^{c} m_{\mu k}^{f} d^{\lambda}(\Gamma^{\mu}, \beta^{k}) \quad [5]$$

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, Γ^{μ} est le vecteur qui représente chaque exemple μ et $d^{\lambda}(\Gamma^{\mu}, \beta^{k})$ est la distance entre l'exemple Γ^{μ} et β^k en accord avec une définition de distance (voir 5.1) et que nous écrirons 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 $m_{\mu k}$:

$$m_{\mu k} = \frac{d_{\mu k}^{\lambda/(f-1)}}{\sum_{j=1}^{c} d_{\mu j}^{\lambda/(f-1)}} \; ; \; \beta^k = \frac{\sum_{\mu=1}^{P} m_{\mu k}^f \Gamma^\mu}{\sum_{\mu=1}^{P} m_{\mu k}^f}; \; \mu = 1, \cdots, P; \; k = 1, \cdots, c \quad [6]$$

L'intérêt d'utiliser *fuzzy k-means* dans le cadre de la classification thématique de courriers électroniques (Kessler *et al.*, 2004a; Kessler *et al.*, 2004b) 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.

5.1. *Distance entre vecteurs*

Afin d'effectuer la classification, nous calculons la distance entre les vecteurs et les centroïdes. Nous avons utilisé, pour cela, la distance de Minkowski $d^\lambda(a,b) = \left(\sum_i ||a_i - b_i||^\lambda\right)^{1/\lambda}$. Nous avons fait une implantation de l'algorithme avec $\lambda = 1$ (distance de *Manhattan*), cependant les résultats obtenus étant décevants, nous avons utilisé $\lambda = 2$ (distance euclidienne) avec $d^2 = (\Gamma^\mu, \beta^k) \leftarrow \left\|\Gamma^\mu - \beta^k\right\|$. La distance de Manhattan permet principalement de faire la différence entre la présence ou l'absence d'un terme i dans un courriel μ, et la distance euclidienne apporte en plus les poids de chaque terme, ce qui permet de mieux classer les exemples.

5.2. *Initialisation aléatoire ou semi-supervisée ?*

K-means et *fuzzy k-means* sont des algorithmes performants mais fortement dépendants de l'initialisation (Fred *et al.*, 2003). Nous étions donc confrontés au problème de l'initialisation des centroïdes β^k. Nous avons d'abord testé la méthode avec des initialisations aléatoires, mais l'erreur d'apprentissage, ϵ_a, était autour de 25 % dans le meilleur des cas (voir figure 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 nos centroïdes. Nous avons fait une étude de cette initialisation semi-supervisée. Sur la figure 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 % et 20 % pour obtenir ϵ_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 3 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.

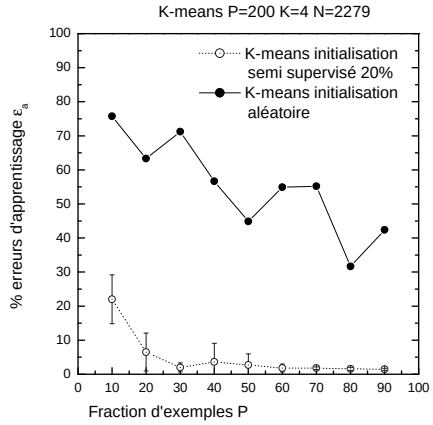

Figure 2. *Comparaison entre l'initialisation aléatoire et l'initialisation semi-supervisée. Dans tous les cas, les moyennes sont calculées sur 10 tirages aléatoires*

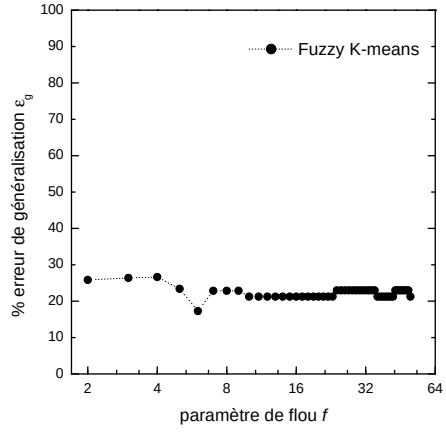

Figure 3. *Incidence du log_2(paramètre flou f) sur l'erreur de généralisation ϵ_g. On a P = 160 courriels en apprentissage et* 40 *en test*

6. 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 al.*, 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 de recherche documentaire 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 radicalisation.

Etant donné la faible quantité de mots (11 mots en moyenne) contenues dans les messages electroniques, des tests préliminaires concernant leur classification, nous ont montré que les n-grammes de lettres ont des performances supérieures à celles de n-grammes de mots. Pour tenir en compte de ces observations, nous avons décidé donc d'utiliser les n-grammes de lettres à la place des n-grammes de mots comme termes de la matrice Γ. Nous avons donc effectué une implantation différente de la partie prétraitement afin de pouvoir construire une matrice basée sur les fréquences des n-grammes de lettres et non sur la fréquence des mots, étant $n = 2$. Nous effectuons pour cela un découpage de chacun de nos messages en n-grammes avec ou sans lemmatisation afin de pouvoir effectuer un comparatif entre les deux méthodes ainsi qu'en fonction de la racinisation des termes.

7. Apprentissage supervisé

Les machines à vecteurs de support, proposées par Vapnik (Vapnik, 1995) ont été utilisées avec succès dans plusieurs tâches d'apprentissage. Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel. La méthode repose sur les idées suivantes :

– les données sont projetées dans un espace de grande dimension par une transformation basée sur un noyau linéaire, polynomial ou gaussien comme le montre la figure 4 ;

– dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui maximisent la marge (distance entre les classes) ;

– les hyperplans peuvent être déterminés au moyen d'un nombre de points limités qui seront appelés les " vecteurs de support " ;

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). 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; Joachims, 1999), mais toujours en utilisant des corpus bien rédigés (des articles journalistiques, scientifiques...). Nous avons décidé de tester leur robustesse sur un type de corpus particulièrement bruité : le courrier électronique.

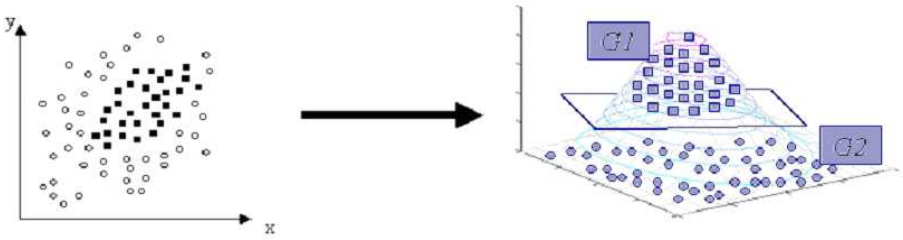

Figure 4. *À gauche, nous avons un ensemble de données non linéairement séparables dans un espace bidimensionel (x,y). À droite, au moyen d'une fonction noyau, ces données sont projetées dans un espace tridimensionnel, où un hyperplan sépare correctement les deux classes $G1$ et $G2$*

Implantation

Nous avons testé plusieurs implantations différentes des machines à vecteurs de support (Lia_sct (Bechet *et al.*, 2000), SVMTorch[12], Winsvm[13], M-SVM[14]) afin de pouvoir sélectionner la plus efficace. Nous avons finalement décidé d'utiliser une implantation de l'algorithme de Collobert (Collobert *et al.*, 2000), SVMTorch, qui permet une approche multiclasse des problèmes de classification. Celle-ci utilise le principe *One-against-the-Rest*, où chaque classe est comparée à l'ensemble des autres afin de trouver un hyperplan séparateur. On utilise pour cela une fonction appelée noyau, qui permet de projeter les données dans un espace de grande dimension de la façon suivante : sous les conditions de Mercer (Vapnik, 1995), le produit scalaire dans le nouvel espace peut être réécrit au moyen d'une fonction noyau $K(a, b)$ telle que :

$$K(a, b) = (\Phi(a)) \bullet (\Phi(b)) \qquad [7]$$

12. *http ://www.idiap.ch/*
13. *http ://liama.ia.ac.cn/PersonalPage/lbchen/winsvm.htm*
14. *http ://www.loria.fr/ guermeur/*

SVMTorch permet de tester plusieurs types de fonctions noyaux :

– un noyau simple (polynôme de premier degré) ;

– un noyau polynomial de degré d $(a, b) \rightarrow (a \cdot b)^d$;

– gaussiennes à base radiale (FBR) $(a, b) \rightarrow \exp\left(-\frac{||a-b||^2}{2\sigma 2}\right)$;

– fonctions d'activation sigmoïdales $(a, b) \rightarrow \tanh(sa \cdot b + r)$;

Les résultats des tests que nous allons présenter dans la section 9, ont été obtenus à l'aide d'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.

8. La méthode hybride

Nous avons 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. La figure 5 montre la chaîne de traitement complète.

9. Expériences

Nous avons travaillé avec des corpus de P = {200, 500, 1 000, 2 000} courriels ayant k = 4 classes parmi {**football, jeux de rôles, cinéma, ornithologie**}. Chacun des tests a été effectué à 50 ou 100 tirages aléatoires, dans le cas de n-grammes ou de mots, respectivement. *Fuzzy k-means* nécessite un calibrage du paramètre de flou f. Nous avons decidé de garder la valeur de f qui obtient une erreur de généralisation, sur les ensembles de test, la plus basse. La figure 3 présente l'incidence du paramètre *f*. La figure 2 présente une comparaison entre une initialisation aléatoire et une initialisation semi-supervisée.

La figure 6 présente les résultats obtenus sur un corpus de P = 500 courriels. À gauche, nous montrons l'erreur d'apprentissage ϵ_a avec une initialisation semi-supervisée pour *k-means*. À droite, nous montrons l'erreur en généralisation ϵ_g de SVM avec un apprentissage supervisé. Bien sûr, 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 nécessitent une classification préalable des exemples de l'ensemble d'apprentissage, ce qui n'est toujours pas disponible.

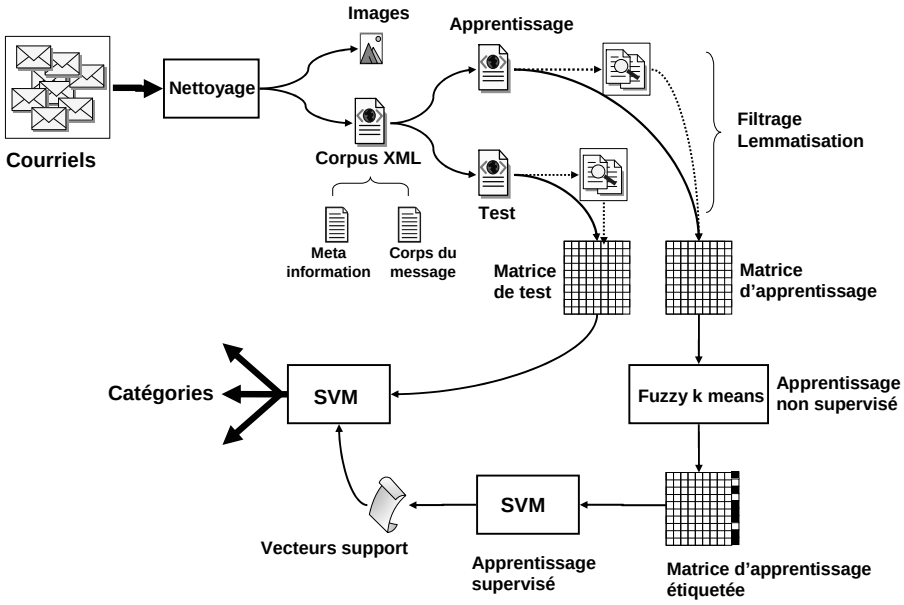

Figure 5. *Méthode hybride, chaîne de traitement complète : classification d'un corpus de courriels en k catégories*

La figure 7 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 avec nos corpus et sensiblement comparables à ceux obtenus avec un modèle unigramme de mots. Nous montrons l'erreur en généralisation ϵ_g pour la méthode hybride.

Les figures 8 et 9 comparent les résultats de la méthode hybride et des SVM supervisées sur des corpus de $P = \{200, 500\}$ et de P = {1 000, 2 000} courriels respectivement. Dans le cas hybride, nous avons combiné un apprentissage non supervisé par *k-means* (initialisation semi-supervisée de $0.05P$ ou $0.2P$ courriels) et supervisé pour SVM. Nous constatons que la performance ne se détériore pas en augmentant la taille du corpus. Il est possible de voir aussi que les performances en généralisation de la méthode hybride sont très proches de celles des SVM en apprentissage supervisé. Notons que dans toutes les courbes des figures 8 et 9 l'unité de base est un unigramme de mot.

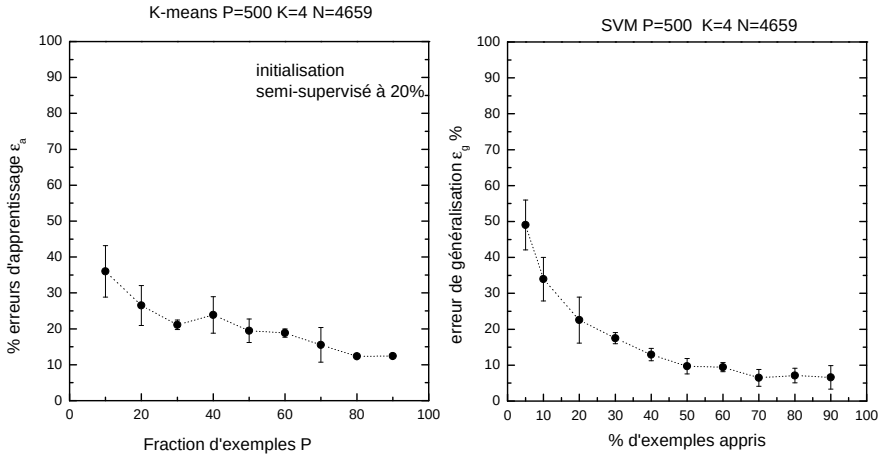

Figure 6. *Erreur d'apprentissage ϵ_a pour k-means à gauche et de généralisation ϵ_g SVM à droite, P = 500 courriels. k = 4 classes, N = dimension des matrices*

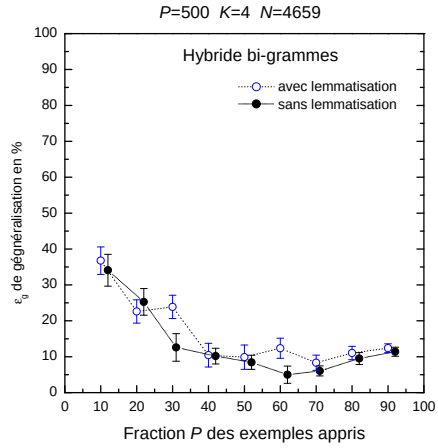

Figure 7. *Erreur de généralisation pour la méthode hybride avec bigrammes de lettres (50 tirages aléatoires) sur un corpus de P = 500 courriels. Le fait de ne pas lemmatiser améliore sensiblement les résultats. k = 4 classes, N dimension des matrices*

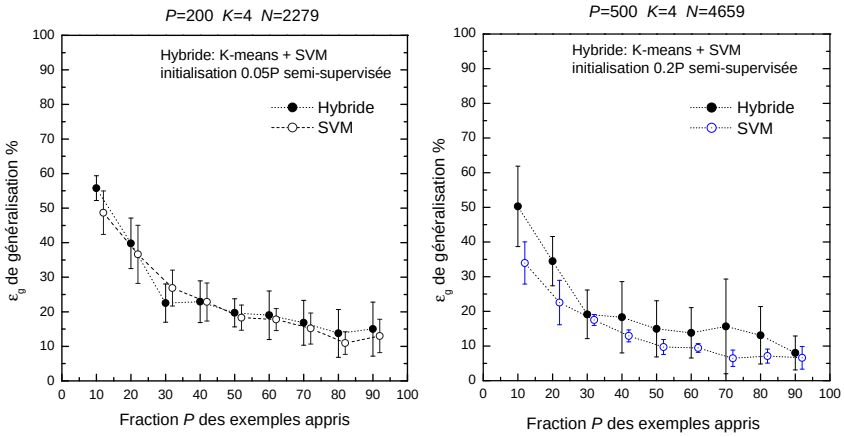

Figure 8. *Méthode hybride vs. SVM, P = 200 courriels à gauche et 500 courriels à droite. 100 tirages aléatoires. k = 4 classes, N = dimension des matrices*

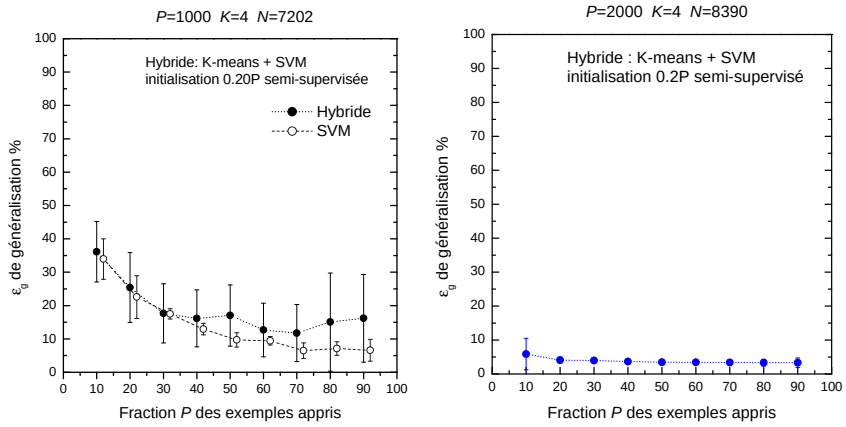

Figure 9. *Méthode hybride vs. SVM. À gauche P = 1 000, à droite 2 000 courriels. L'erreur est négligeable au délà de 20 % des exemples appris. Par souci de clarté seulement la courbe de la méthode hybride est presentée, celle-ci se confondant avec la courbe SVM. k = 4 classes, N = dimension des matrices*

Discussion

Nous avons étudié en détail 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 ci-après, deux exemples de messages qui n'ont pas été bien classés par notre système. Le premier est trop court et trop vague pour permettre une catégorisation adéquate, tandis que 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 de cette tâche, qui n'est pas évidente, même pour les personnes.

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

```
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: <c7Oti1+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
Reply-To: <Shadowrun-france@yahoogroupes.fr>

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
Reply-To: france-foot@yahoogroupes.fr

Nantes  (L1)    1-1     Rennes  (L1)
Ôh purée, j'y etais et une fois de plus j'ai été déçu..
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...
```

---

10. Conclusion et perspectives

La tâche de classification de courriels est assez 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 prétraitement (filtrage, traduction, lemmatisation) afin de représenter les courriels dans un modèle vectoriel. Nos travaux concernant les dictionnaires avec ou sans accents ont permis d'accroître les performances du filtrage. À partir de cette représentation des données, nous avons réalisé une étude de différentes méthodes d'apprentissage automatique. Cette étude

nous a permis de mieux connaître les caractéristiques de ces méthodes et leur comportement sur les données que nous avons collectées. Ainsi, l'apprentissage supervisé permet de mieux classer des nouveaux courriels mais demande une classification 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. De plus, cette méthode n'a pas besoin de données étiquetées. La méthode hybride, qui permet de combiner 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 intéressants : avec 20 % d'exemples appris (400 courriels), l'erreur est inférieure à 5 % dans le cas d'un ensemble d'apprentissage de 2 000 courriels.

Deux méthodes hybrides, découpage en mots et n-grammes de lettres, ont été développées et testées. Les résultats avec les bigrammes semblent être similaires à ceux avec des mots. 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, sont aussi 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. Le noyau des SVM le plus adapté a eté celui linéaire. La prochaine étape consistera à optimiser les SVM afin de pouvoir améliorer les performances globales du système. Ainsi, une implantation des machines à vecteurs de support selon l'algorithme DDAG (Kijsirikul *et al.*, 2002) permettrait d'éliminer les régions inclassifiables et d'obtenir une meilleure classification des nouvelles données. De même, une combinaison de classifieurs (Grilheres *et al.*, 2004) pourrait permettre d'améliorer nos résultats.

11. Bibliographie

Beauregard S., Génération de texte dans le cadre d'un système de réponse automatique à des courriels., Thèse de doctorat, Université de Montréal, Québec, Canada, 2001.

Bechet F., Nasr A., Genet F., "Tagging Unknown Proper Names Using Decision Trees", *38th Annual Meeting of the Association for Computational Linguistics*, Hong-Kong, p. 77-84, Octobre, 2000.

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

Cohen W. W., "Learning Rules that Classify Email", *AAAI Spring Simposium on Machine Learning in Information Access*, Minneapolis, Min., USA, p. 18-25, 1996a.

Cohen W. W., "Learning to Classify English Text with ILP Methods", *in* L. De Raedt (ed.), *Advances in Inductive Logic Programming*, IOS Press, p. 124-143, 1996b.

Collobert R., Bengio S., On the convergence of SVMTorch, an algorithm for large-scale regression problems, Technical Report n° IDIAP-RR 00-24, IDIAP, Martigny, Switzerland, 2000.

Damashek M., "A Gauging Similarity with N-Grams : Language-Independent Categorization of Text", *Science*, vol. 267, p. 843-848, 1995.

deGruijter J., McBratney A., "A modified fuzzy k means for predictive classification", *in* H. Bock (ed.), *Classification and Related Methods of Data Analysis*, Elsevier Science, Amsterdam, p. 97-104, 1988.

Dunning T., Statistical Identification of Languages - Technical Report MCCS 94-273, Algorithms - computer science, Computing Research Laboratory, 1994.

Fred A., Jain A., "Robust data clustering", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 128-133, 2003.

Grefenstette G., "Comparing Two Language Identification Schemes", *in* JADT'96 (ed.), *3rd International Conference on the Statistical Analysis of Textual Data*, Rome, Italia, p. 263-268, 1996.

Grilheres B., Brunessaux S., Leray P., "Combining classifiers for harmful document filtering", *Recherche d'Information Assistée par Ordinateur*, Avignon, p. 173-185, 2004.

Jake D. Brutlag C. M., "Challenges of the Email Domain for Text Classification", *Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, p. 103-110, 2000.

Jalam R., Chauchat J.-H., « Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques », *in* J. 2002 (ed.), *6^{e} Journées internationales d'Analyse statistique des Données Textuelles*, St-Malo, p. 77-84, Octobre, 2002.

Joachims T., "Text Categorization with Support Vector Machines : Learning with Many Relevant Features", *10th European Conference on Machine Learning*, Chemnitz, Germany, p. 137-142, 1998.

Joachims T., "Transductive Inference for Text Classification using Support Vector Machines", *16th International Conference on Machine Learning*, San Francisco, USA, p. 200-209, 1999.

Kessler R., Torres-Moreno J. M., El-Bèze M., « Classification thématique de courriels », *Journée ATALA sur les nouvelles formes de communication écrite*, ATALA, http ://www.up.univ-mrs.fr/veronis/je-nfce/resumes.html, Paris, Juin, 2004a.

Kessler R., Torres-Moreno J. M., El-Bèze M., « Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé », *VSST 2004*, vol. B, Toulouse, p. 493-504, 2004b.

Kijsirikul B., Ussivakul N., "Multiclass Support Vector Machines Using Adaptive Directed Acyclic Graph", *International Joint Conference on Neural Networks*, vol. 1, Lausanne, p. 980-985, 2002.

Kiritchenko S., Matwin S., "Email Classification with Co-Training", *Conference of the Centre for Advanced Studies on Collaborative research*, Toronto, Ontario, Canada, 2001.

Kosseim L., Beauregard S., Lapalme G., "Using Information Extraction and Natural Language Generation to Answer E-Mail", *Lecture Notes in Computer Science*, vol. 1959, p. 152-163, 2001a.

Kosseim L., Lapalme G., « Critères de sélection d'une approche pour le suivi automatique du courriel », *8^{e} conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN)*, p. 357-371, 2001b.

Lebart L., « Validité des visualisations de données textuelles », *6eme International Conference on the Statistical Analysis of Textual Data*, France, p. 708-715, 2004.

Lewis D. D., Ringuette M., "A comparison of two learning algorithms for text categorization", *SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, p. 81-93, 1994.

Manning C. D., Schütze H., *Foundantions of Statistical Natural Language Processing*, The MIT Press, 2000.

Miller E., Shen D., Liu J., Nicholas C., "Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System", *Journal of Digital Information*, vol. 1, n° 5, p. http ://jodi.tamu.edu/Articles/v01/i05/Miller/, 2000.

Namer F., « Un analyseur flexionnel du français à base de règles », *TAL*, vol. 41, n° 2, p. 247-523, 2000.

Sahami M., Using Machine Learning to Improve Information Access., PhD thesis, Computer Science Department, Stanford University, 1999.

Torres-Moreno J. M., Velázquez-Morales P., Meunier J.-G., « Classphères : un réseau incrémental pour l'apprentissage non supervisé appliqué à la classification de textes », *JADT 2000*, vol. 2, M. Rajman & J.-C. Chappelier éd., Suisse, p. 365-372, 2000.

Torres-Moreno J. M., Velázquez-Morales P., Meunier J.-G., « Cortex : un algorithme pour la condensation automatique des textes », *ARCo*, vol. 2, Hermès Science France, p. 365-366, 2001.

Torres-Moreno J. M., Velázquez-Morales P., Meunier J.-G., « Condensés de textes par des méthodes numériques », *JADT*, vol. 2, France, p. 723-734, 2002.

Vapnik V., *The Nature of statistical Learning Theory (second ed.)*, Springer, 1995.

Vinot R., Grabar N., Valette M., « Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'Internet », *TALN*, France, p. 275-284, 2003.