Classification thématique de courriels par une méthode hybride d'apprentissage

Rémy Kessler, Juan Manuel Torres-Moreno et Marc El-Bèze
Laboratoire d'Informatique d'Avignon - Université d'Avignon et des Pays de Vaucluse
339 chemin des Meinajaries, Agroparc, BP 1228, 84911 AVIGNON Cedex 9, FRANCE
e-mail : {kessler, torres, elbeze}@lia.univ-avignon.fr

9 juin 2004

Résumé

Les nouvelles formes de communication écrite (courrier électronique, forums, chats, SMS, etc.) présentent des défis considérables pour leur traitement. Nous présentons des recherches destinées à créer des outils et des ressources génériques pour la classification de courriels. La capacité d'une entreprise de gérer efficacement, rapidement et à moindre coût, ces flux d'informations devient un enjeu majeur pour la satisfaction des clients. Ceci nécessite, en particulier, de disposer d'outils informatiques permettant notamment : le routage pour acheminer les courriels vers le destinataire concerné et l'automatisation de réponses. Nous nous attachons à traiter dans cette étude des problèmes posés par le routage précis de courriels : après un processus puissant 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. Nous avons trouvé par ailleurs une initialisation semi-supervisée qui optimise l'apprentissage non supervisé. Lors des tests préliminaires, nous avons obtenu bonnes performances sur des corpus réalistes.

Mots-clés : Apprentissage automatique, machines à vecteurs support (SVM), fuzzy k-means, classification de textes, routage automatique de courriels.

1 Introduction

Les nouvelles formes de communication écrite posent des défis considérables aux systèmes de traitement automatique de la langue car on observe des phénomènes linguistiques bien particuliers comme les émoticônes, 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 saisies imposées par le matériel (terminal mobile, téléphone, etc.). Dans cet article, nous désignons par **phonécriture**

ou **phonécrit** toute forme écrite qui utilise un type d'écriture phonétique sans contraintes ou avec des règles établies par l'usage (par exemple *kdo* à la place de *cadeau, a+* pour *à plus, 10ko* pour *dictionnaire*, etc.). Le traitement automatique des courriels est extrêmement difficile à cause de son caractère imprévisible [1, 8] : des textes trop courts (moyenne autour de 11,02 mots par courriel), régis par une syntaxe pauvre ou mal orthographiés. 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), et en particulier les méthodes fondées sur les réseaux de neurones, 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, combinant techniques probabilistes et *support vector machines* (SVM). 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 incomplèts et/ou très bruitées.

2 Définition du problème

On se place dans le cas où une boîte aux lettres reçoit un grand nombre de courriels correspondant à plusieurs thématiques et qu'une personne doit les lires et les rediriger vers le service concerné (les courriels concernant un type de problèmes techniques vers le service technique, ceux pour le service après vente seront redirigés vers ce dernier, etc.). Le système permettrait donc de faire cette partie du travail automatiquement. Après une recherche sur Internet, il s'est averé difficile de trouver des corpus de courriels en français (des corpus anglais existent cependant pour de la classification de *spam*). Nous avons donc décidé de créér une adresse électronique et de l'abonner à diverses listes de diffusion [1] où *newsletters* [2] de thèmes variés. Ainsi, un corpus réaliste a été crée. Nous avons donc créé différents corpus de plusieurs tailles et avec des caractéristiques particulières comme le montre l'exemple le tableau 1.

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 lin-

[1] Football, jeux de rôles, ornithologie, cinéma, jeux vidéo, poème, humour, etc.

[2] Sécurité informatique, journaux, matériel informatique, etc.

Statistiques du corpus		
Nb. total de courriels	$P = 1000$	
Nb. total de mots bruts	10016	
Nb. de courriels avec pièce jointe	150	15,00%
Nb. courriels avec sujet long ($>$ 3 mots)	622	62,20%
Nb. courriels avec sujet court ($<$ 3 mots)	378	37,80%
Nb. de courriels court ($<$ 10 mots)	665	66,50%
Nb. de courriels long ($>$ 10 mots)	335	33,50%
Taille moyenne d'un courriel en mots	11,02	
Nb. d'auteurs différents	247	
Auteurs ayant envoyé moins de 2 courriels	69	27,93%
Auteurs ayant envoyé entre 2 et 5 courriels	125	50,60%
Auteurs ayant envoyé entre 5 et 10 courriels	23	9,31%
Auteurs ayant envoyé plus de 10 courriels	30	12,14%

TAB. 1 – *Statistiques générées sur le corpus de $P = 1000$ courriels utilisé dans cette étude.*

guistique, s'avère performante et rapide [9]. Ces méthodes ont par ailleurs la propriété d'être sensiblement indépendantes de la langue [3]. La première étape consiste à nettoyer le corpus afin de séparer l'en-tête, le corps et la pièce jointe du courrier électronique. Notre classification s'effectue pour l'instant uniquement sur le corps du message. L'en-tête représente une meta-information qui peut aussi être très utile pour la tâche de classification. Par la suite, nous réalisons un pré-traitement où des processus puissants de filtrage et de lemmatisation sont déclenchés afin de réduire la dimensionalité des matrices [11, 12, 13]. La lemmatisation (même si elle est plus coûteuse en temps) et non un simple *stemming* s'avère plus performante pour le français, qui est une langue romaine à flexion forte [6]. Au cours de cette phase, nous effectuons différents traitements que nous détaillerons par la suite en vue de réduire cette dimensionalité. Nous obtenons donc à la fin de ce pré-traitement une matrice ξ tel que (ξ_j^μ) avec $\mu = 1 \cdots P, j = 1 \cdots N$ de P exemples et N termes différents qui sera, par la suite, divisée aléatoirement en sous-ensembles d'apprentissage et de généralisation, puis traitée par les méthodes d'apprentissage. Nous avons décidé d'utiliser les algorithmes k-means et fuzzy k-means pour l'apprentissage non supervisé et SVM pour l'apprentissage supervisé. La figure 1 illustre l'ensemble des opérations.

3.1 Pré-traitement

La première partie du pré-traitement a consisté à identifier dans le corpus chaque courriel différent, séparer le corps du message des pièces jointes. Cette

[3] Pour l'instant nos tests sont limités au français.

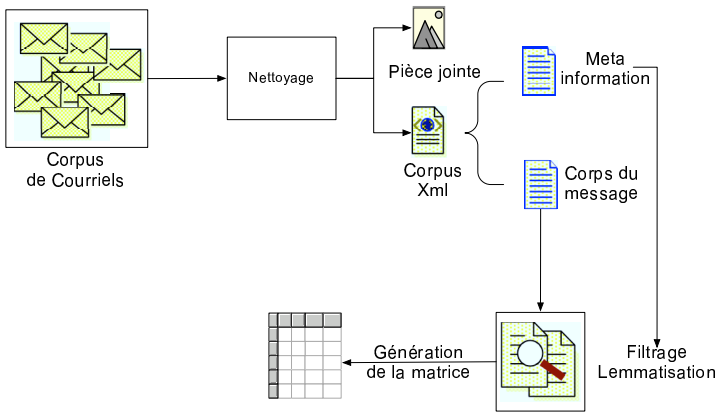

FIG. 1 – schéma du pré-traitement global

première étape génère un fichier présenté ci-dessous, au format XML à la structure simple, ceci afin d'en faciliter son parcours par la suite.

```
 <?xml version="1.0" encoding="UTF-8"?>
<Corpus>
    <Mail>
      <Head>
        <From>
          <Name>
            Nom de l'expéditeur du courriel
          </Name>
          <Mail_adresse>
            adresse électronique de l'expéditeur du courriel
          </Mail_adresse>
        </From>
        <Date>
          Date d'envoi du message
        </Date>
        <Destinataire>
          <To>
            <Name>
              Nom du destinataire du courriel
            </Name>
            <Mail_adresse>
              adresse du destinataire du courriel
            </Mail_adresse>
          </To>
        </Destinataire>
        <Subject>
          sujet du message
        </Subject>
      </Head>
      <Body>
        contenu du message
      </Body>
    </Mail>
    <Mail>
      ...
    </Mail>
</Corpus>
```

Cette étape nous permet par ailleurs de générer les statistiques du tableau 1. Une fois cette tâche réalisée, un second processus se charge d'effectuer les traitements de filtrage. Ainsi on supprime la micro-publicité (*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 de publicités ajoutées au bas des courriels par les fournisseurs de service de messagerie éléctronique comme le montrent les exemples suivants.

```
_____________________________________________________________________
Envie de discuter en "live" avec vos amis ? Télécharger MSN Messenger
http://www.ifrance.com/_reloc/m la 1ère messagerie instantanée de France

___[ Pub ]_____________________________________________________________
Inscrivez-vous gratuitement sur Tandaime, Le site de rencontres !
http://rencontre.rencontres.com/index.php?origine=4

_____________________________________________________________________
Yahoo! Mail : votre e-mail personnel et gratuit qui vous suit partout !
Créez votre Yahoo! Mail

Dialoguez en direct avec vos amis grâce à Yahoo! Messenger !
```

Nous nous sommes attachés à supprimer la publicité générique des courriels, celle-ci étant généralement précédée d'une ligne composée de la facon suivante "___[Pub]_____", "_____" ou encore "**********". La particularité de ces lignes a permis de les enlever sans risque de pertes d'informations au niveau du corps du message. Nous avons par la suite supprimé la micro-publicité 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 [4] 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 (par exemple :) → sourire, A+ → à plus tard, @2m1 → à demain).

Par la suite, nous appliquons les processus classiques de traitement de la langue :

Filtrage : Le texte original comporte N_P mots qui peuvent être des mots fonctionnels ou outils (articles, prépositions, adjectifs, adverbes ...), des noms ou des verbes conjugués, mais aussi des mots composés qui représentent souvent un concept bien spécifique. Tous ces mots peuvent être répétés ou non. Il est important de définir si on travaille sur des formes fléchies ou des formes de base. C'est pourquoi on emploie plutôt la notion de *terme* pour désigner un mot plus abstrait. Pour réduire la complexité du texte, différents filtrages du lexique sont effectués : la suppression des verbes et des mots fonctionnels (*être, avoir, pouvoir, falloir* ...), des expressions courantes (*par exemple, c'est-à-dire, chacun de* ...), de chiffres (numériques et/ou textuelles) et des symboles comme <$>, <#>,<*>, etc.

http ://www.mobimelpro.com/portail/fr/my/dictionnaire_sms.asp

[4] http ://www.mobilou.org/10kosms.htm
http ://www.affection.org/chat/dico.html

Lemmatisation : Il est utile de lemmatiser les verbes des langues à fort taux de flexion (comme le sont les langues romanes). Ce traitement entraîne une réduction importante du lexique. La lemmatisation simple consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et/ou féminins au masculin singulier [5] avant de leur associer nombre d'occurrences. Ce processus permet de diminuer la *malédiction dimensionnelle* [6] 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 vectoriel. D'autres mécanismes de réduction du lexique sont aussi déclenchés. Ainsi, les mots composés sont repérés à l'aide d'un dictionnaire, puis transformés en un terme unique lemmatisé [7].

Le pré-traitement transforme donc le corpus initial en un ensemble de vecteurs $P \times N$ avec P le nombre de courriels et N le lexique de termes différents où chaque valeur représente une fréquence. Nous obtenons donc une matrice fréquentielle, $\mathbf{\Gamma} = (\overrightarrow{\Gamma}^{\mu}); \mu = 1, \cdots, P$ où chaque composante $\overrightarrow{\Gamma^{\mu}} = (\Gamma_1^{\mu}, \Gamma_2^{\mu}, \cdots, \Gamma_N^{\mu})$ contient la fréquence Γ_i^{μ} du terme i dans un courriel μ :

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

3.2 Réduction de la taille de la matrice

Nous avons émis l'hypothèse que la suppression d'un terme i présent uniquement pour un courriel μ (même celui-ci ayant une fréquence $\Gamma_i^{\mu} >> 1$) devrait améliorer sensiblement les performances des algorithmes. En effet si un terme n'est présent que dans un seul courriel, celui-ci n'est pas représentatif de sa catégorie k et peut donc être supprimé [8]. Nous avons procédé ainsi afin de réduire le bruit dans la matrice. Nous obtenons donc une nouvelle matrice γ de dimension réduite N_r tel que $N_r < N$, comme le montre le tableau 2.

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

[6] *The curse of dimensionality*.

[7] *pomme de terre* et *pommes de terre* deviennent ainsi **pomme_de_terre**.

[8] Il est possible de se demander si un terme j présent que dans un faible nombre de courriels pourrait aussi être éliminé. Des tests sont en cours.

k classes	P exemples	N_b termes bruts	N termes	N_r termes réduits	Compression ρ en %
3	100	1757	1658	1385	78.82 %
4	100	2130	1902	1585	74.33 %
3	200	2984	2713	2512	84.82 %
4	200	3228	2668	2279	70.68 %
3	500	5550	4448	4030	72.62 %
4	500	5304	4659	4098	77.26 %
4	1000	10016	7893	7202	71.9 %

TAB. 2 – *Résumé de la compression de matrice par corpus*

Le tableau 2 indique le pourcentage de réduction obtenu pour chaque corpus. Si l'on défini ρ comme le rapport entre le lexique brut et le lexique réduit :

$$\rho = \frac{N_r}{N_b}$$

On constate que $N_r < N < N_b$ et que $N_r \approx 0,75N_b$. Nos tests montrent par ailleurs une légère augmentation des performances en ce qui concerne l'apprentissage par fuzzy k-means/k-means et une diminution significative du temps de traitement pour les deux méthodes.

3.3 Observation de la matrice

La figure 2 présente une répartition des termes en fonction des courriels. Les classes ont été mélangées de façon aléatoire lors de la création, ce qui explique la division de certaines classes. L'axe des ordonnées est la liste des termes pour le corpus tandis que l'axe des abscisses représente la liste des courriels. Sur ce graphique, on observe que la categorie 4 est divisée en 2 parties (courriels 1 à 25 puis courriels 100 à 125). La densité des 1000 premiers termes redevient importante entre le 100ème et le 125ème termes, mais ailleurs elle reste assez faible. Ceci est observable aussi pour la catégorie 2 (entre le courriel 25 et 50 puis entre le courriel 125 et 150) ou l'on remarque très bien l'absence de termes des catégories 3 et 4 dans le second morceau (faible densité des termes 1500 à 2500). On observe par ailleurs que lors de l'apparition d'une nouvelle catégorie, les nouveaux termes sont en forte densité dès le début de celle-ci.

3.4 Apprentissage non supervisé avec fuzzy k-means et k-means

L'algorithme Fuzzy k-means [2, 5] 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,

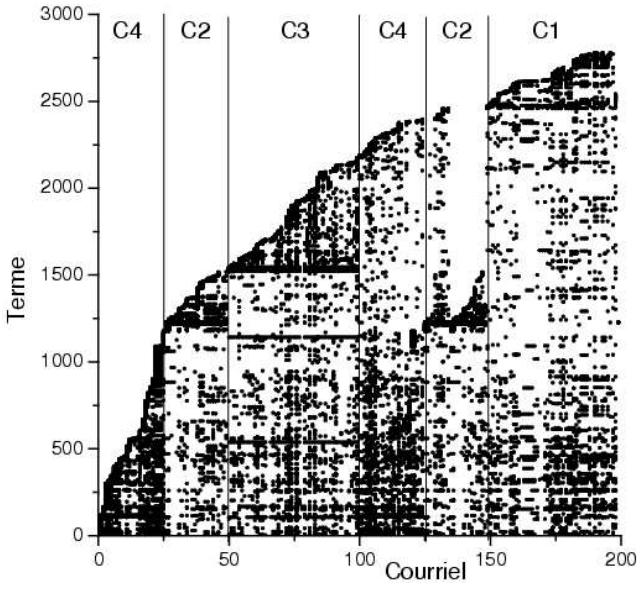

FIG. 2 – Représentation des termes en fonction des courriels.

où chaque exemple appartient à une seule classe (partition dure). Fuzzy k-means minimise la somme des erreurs quadratiques avec les conditions suivantes :

$$\sum_{k=1}^{c} m_{\mu k} = 1; \quad \mu = 1, 2, \cdots, P; \tag{2}$$

$$\sum_{k=1}^{c} m_{\mu k} > 0; \quad k = 1, 2, \cdots, c \tag{3}$$

$$m_{\mu k} \in [0, 1] \quad \mu = 1, 2, \cdots, P; \quad k = 1, \cdots, c \tag{4}$$

On définit alors la fonction objective :

$$J = \sum_{\mu=1}^{P} \sum_{k=1}^{c} m_{\mu k}^{f} d^{\lambda}(\xi^{\mu}, c_k) \tag{5}$$

où P est le nombre de données dont on dispose, c est le nombre de classes désiré, c_k est le vecteur qui représente le centroïde (barycentre) de la classe k, ξ^{μ} est le vecteur qui représente chaque exemple μ et $d^{\lambda}(\xi^{\mu}, c_k)$ est la distance entre l'exemple ξ^{μ} et c_k en accord avec une définition de distance (voir 3.4.1) et que nous allons écrire comme $d_{\mu k}^{\lambda}$ afin d'allèger la notation. f est le paramètre flou, valeur prise dans l'intervalle [2, ∞) qui détermine le degré de flou de la solution finale, contrôlant le degré de recouvrement entre les classes. Avec $f = 1$, la solution devient une partition dure. Si $f \to \infty$ la solution approche le maximum de *fuzzyfication* et toutes les classes risquent de se confondre en une seule. La minimisation

de la fonction objective J fournit la solution pour la fonction d'appartenance $m_{\mu k}$ (4) :

$$m_{\mu k} = \frac{d_{\mu k}^{\lambda/(f-1)}}{\sum_{j=1}^{c} d_{\mu j}^{\lambda/(f-1)}} \quad \mu = 1, 2, \cdots, P;\ k = 1, \cdots, c \tag{6}$$

$$c_k = \frac{\sum_{\mu=1}^{P} m_{\mu k}^{f} \xi}{\sum_{\mu=1}^{P} m_{\mu k}^{f}} \quad k = 1, 2, \cdots, c \tag{7}$$

L'algorithme fuzzy k-means est donc le suivant :

1. Fixer le nombre k de classes $1 < k <$ c ;
2. Fixer une valeur du paramètre flou $f > 2$;
3. Choisir une définition de distance adéquate $d_{\mu k}^{\lambda}$ (voir la sous-section 3.4.1) ;
4. Fixer le critère d'arrêt ε ;
5. Initialiser $m_{\mu k} \leftarrow m_{\mu k}^{0}$ (voir la sous-section 3.4.3) ;
6. A l'itération $it = 1, 2, 3, \cdots$ (re) calculer $c_k^{it} \leftarrow c_k^{it-1}$ en utilisant (7) et $m_{\mu k}^{it-1}$;
7. Re-calculer $m_{\mu k} \leftarrow m_{\mu k}^{it}$ en utilisant (6) et c_k^{it} ;
8. Si $||m_{\mu k}^{it} - m_{\mu k}^{it-1}|| < \varepsilon$ alors stop, sinon retourner à 6.

L'intérêt d'utiliser fuzzy k-means dans le cadre de la classification thématique de courrier électronique correspond à router un courriel vers un destinataire prioritaire (celui avec le degré d'appartenance le plus élevé) et en copie carbone (Cc) ou cachée (Bcc) vers celui (ou ceux) dont le degré d'appartenance dépasse un certain seuil établi à l'avance. Cela peut cependant être réalisé par k-means si l'on prend une distance normalisée inverse entre les centroïdes et le courriel.

3.4.1 Calcul de distance entre les 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 :

$$d^{\lambda}(a, b) = \left(\sum_{i} ||a_i - b_i||^{\lambda} \right)^{1/\lambda} \tag{8}$$

dans un premier temps, nous avons utilisé $\lambda = 2$, ceci correspondant à une distance euclidienne $d2(\xi, c_k) \leftarrow ||\xi - c_k||^2$. Afin d'améliorer nos premiers résultats, nous avons par la suite implémenté une version de notre algorithme avec $\lambda = 1$, aussi appelé distance de *Manhattan*. Cependant, les résultats obtenus étant plus décevants,

nous sommes revenus à la distance euclidienne. Nous attribuons cette baisse au fait que la distance de *Manhattan* permet de faire la différence entre la présence ou l'absence d'un terme i dans un courriel μ sans tenir compte de la fréquence de ce terme. Nous obtenions dès lors des distances plus faibles entre un vecteur et chaque centroïde, ceci augmentant la difficulté de l'attribution de la classe correcte par notre système.

3.4.2 Mesure de la performance

Pour estimer les performances des algorithmes nous avons utilisé les mesures d'erreur suivantes, ε_a étant l'erreur en apprentissage et ε_g, l'erreur en généralisation :

$$\varepsilon_a = \frac{\text{Nb. exemples} \in \gamma_1 \text{bien appris}}{\text{card}\{\gamma_1\}} \quad (9)$$

$$\varepsilon_g = \frac{\text{Nb. exemples} \in \gamma_2 \text{bien classés}}{\text{card}\{\gamma_2\}} \quad (10)$$

où card$\{\gamma\}$ représente le nombre d'elements de l'ensemble γ.

3.4.3 Initialisation aléatoire ou semi-supervisée

On sait que k-means et fuzzy k-means sont des algorithmes performants mais fortement dépendants de l'initialisation [16]. On est donc confronté 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 autour de 25% dans le meilleur des cas (voir figure 6). De même l'erreur en généralisation, ε_g, était toujours assez importante. Ceci est dû au fait que l'algorithme semble piégé dans des minimums 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 6, sont illustrés les résultats que nous avons obtenus sur 10 ensembles d'apprentissage exhaustifs tirés au hasard. 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 $\varepsilon_a <$ 10%.

3.4.4 Incidence du paramètre de flou

Nous avons voulu 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 5 montrent qu'au delà d'une valeur de 10, les variations sur ε_a sont négligeables. Nous faison donc varier f entre 1 et 10 dans nos expériences et prenons le meilleur résultat.

3.5 Machines à support vectoriel

Ces machines, proposées par Vapnik [14], ont été utilisées avec succès dans plusieurs tâches d'apprentissage et sont actuellement en plein essor. 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 3.
- dans cet espace transformé, les classes seront 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 supports.

La complexité d'un classifieur SVM va donc dépendre non pas de la dimension de l'espace des données, mais du nombre de vecteurs supports nécessaires pour réaliser la séparation. Les SVM avaient déjà été appliqués au domaine de la classification du texte dans plusieurs travaux [7, 15], mais toujours en utilisant des corpus bien rédigés (des articles journalistiques, scientifiques...). Nous avons décidé de les utiliser dans ce type de corpus particulier, les courriels.

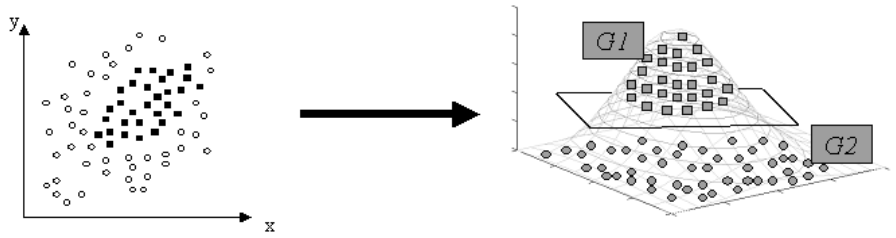

FIG. 3 – En plongeant les données dans un espace de plus grande dimension, on simplifie la tâche de classification

3.5.1 Méthode utilisé

Nous avons testé plusieurs implémentations différentes (lia_sct [9], SVMTorch [10], Winsvm [11], M-SVM [12]) des machines à support vectoriel afin de pouvoir utiliser la plus efficace et qui convenait le mieux a notre problème. Nous avons finalement décidé d'utiliser une implémentation de l'algorithme de Collobert [4], SVMTorch, qui permet une approche multiclasses des problèmes de classification

[9] *lia_sct :a decision-tree based string classifier from F. Béchet*
[10] *http ://www.idiap.ch/*
[11] *http ://liama.ia.ac.cn/PersonalPage/lbchen/winsvm.htm*
[12] *http ://www.loria.fr/ guermeur/*

dans un grand nombre de dimensions. Celle-ci utilise le principe *One-against-the-Rest* (1-v-R), où chaque classe est comparée aux autres afin de trouver l'hyperplan séparateur. On utilise pour cela une fonction noyau qui permet de projeter les données dans un espace de grande dimension de la façon suivante : sous les conditions de Mercer [14], le produit scalaire dans le nouvel espace peut être réecrit au moyen d'une fonction noyau (*kernel*) $K(a, b)$ telle que :

$$K(a, b) = (\Phi(a) \bullet \Phi(b))$$

SVMTorch permet de tester plusieurs types de fonctions noyaux :

- noyau simple (polynôme de premier degré);
- noyau polynomial de degré d $(a, b) \rightarrow (a \cdot b)^d$;
- gaussiennes à base radiale (FBR) $(a, b) \rightarrow \exp\left(-\frac{||a-b||^2}{2\sigma 2}\right)$;
- noyau basé sur une forme particulière de réseau de neurones avec fonctions d'activation sigmoïdales $(a, b) \rightarrow \tanh(sa \cdot b + r)$

Les résultats des tests que nous avons effectués (voir figure 8) ont été trouvés à l'aide d'une fonction à noyau simple. Nous prévoyons aussi de tester d'autres fonctions noyaux afin de savoir si cela influence positivement nos résultats.

3.6 La méthode hybride

Nous avons décidé de combiner les deux méthodes d'apprentissage afin d'avoir les avantages de chacune d'entre elles. En effet, l'apprentissage non supervisé avec k-means avait de bon résultats lors de la phase d'apprentissage mais faisait beaucoup d'erreur en généralisation. D'un autre coté, l'apprentissage avec les SVM est supervisé donc coûteux. En combinant les deux méthodes, nous pensons tirer des avantages des deux, sans les inconvénients.

Ainsi, une fois la réduction de matrice effectuée, nous effectuons un tirage aléatoire de chacun des courriels afin de les séparer en une matrice d'apprentissage γ_1 de P_1 exemples et une matrice de test γ_2 de P_2 exemples sachant $P = P_1 + P_2$. Une fois la séparation terminée, nous effectuons un apprentissage non supervisé avec k-means sur la matrice γ_1. Nous avons modifié notre implémentation de k-means afin que celui-ci, en sortie, nous fournisse γ_1 accompagné de la classe prédite pour chaque courriel. La dernière étape consiste à présenter γ_1 et γ_2 aux machines à support vectoriel, celles-ci pouvant dès lors effectuer un apprentissage supervisé sur γ_1 et donc sur les résultats obtenus par k-means. Le schéma 4 résume la chaîne de traitement au complet.

Les premiers résultats en généralisation obtenus ont montré une faiblesse au niveau de l'écart type. En effet ceux-ci pouvaient être très bons ou beaucoup plus moyens. L'observation en détail des résultats moyens a montré que ceux-ci étaient dus a un apprentissage de mauvaise qualité avec k-means, suite à un échantillonage mauvais lors de l'initialisation. Afin de corriger ce problème, nous avons ajouté un test permettant de refaire une initialisation lorsque les résultats de l'apprentissage

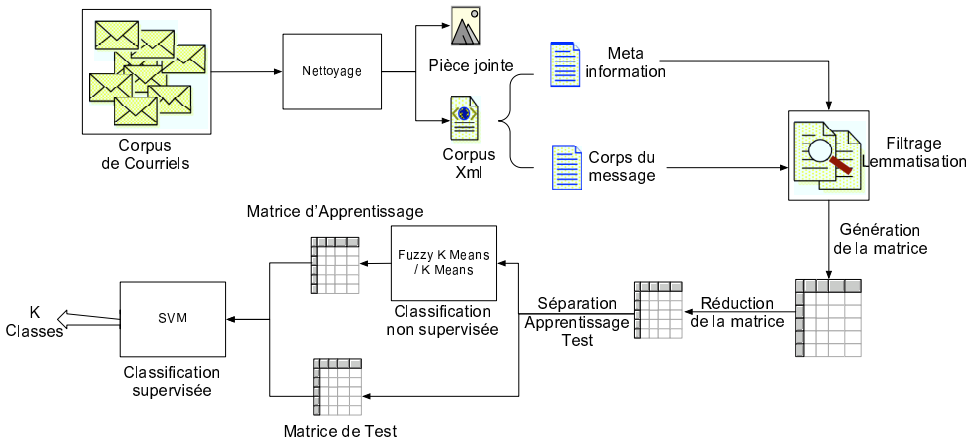

FIG. 4 – Méthode hybride : chaîne de traitement complète.

étaient inférieurs à un certain seuil. Les figures 9 et 10 présentent les résultats obtenus après cette modification.

4 Expériences

Nous avons travaillé avec des corpus de $P = \{100, 200, 500, 1000\}$ courriels ayant $k = \{3, 4\}$ classes parmi {football, jeux de rôles, cinéma, ornithologie}. Les résultats ci dessous présentent l'incidence du paramètre flou ainsi que l'incidence de l'initialisation aléatoire ou semi-supervisée. Chacun de ces tests a été effectué à 10 reprises avec un corpus de $P = 200$ courriels et $k = 4$ catégories différentes et un tirage aléatoire.

On remarque sur la figure 5 qu'au delà de $f = 10$, l'incidence du paramètre flou sur les résultats devient négligeable . La figure 6 présente une comparaison entre une initialisation aléatoire et une intialisation semi supervisée où l'on prend une petite partie P_{ini} de l'ensemble d'apprentissage γ (ici $P_{ini} = 0,2P$) afin d'avoir des centroïdes initiaux de meilleure qualité. Nous constatons une très nette amélioration des résultats avec l'initialisation semi-supervisée.

Comme nous l'avons mentionné en 3.4.3, l'initialisation de $k-$means est essentielle. Nous avons effectué des tests sur l'apprentissage non supervisé afin de connaître les performances de notre système, sur différentes tailles de corpus, avec initialisation aléatoire ou semi supervisée, et différentes méthodes de calcul de distance. Nous présentons ici uniquement les résultats obtenus avec une initialisation semi-supervisée sur les différentes tailles de corpus et $k = 4$ classes différentes :

Les tests suivants ont porté sur l'apprentissage supervisé avec les SVM. Les corpus utilisés sont les mêmes, à savoir $P = 200$ et $P = 500$ courriels avec $k = 4$ classes différentes. Les graphiques 9 et 10 montrent les résultats de la méthode hybride, combinant un apprentissage par la méthode non supervisée (avec initialisation semi supervisée de 20% des courriels) et supervisée par les SVMs, ceux-ci

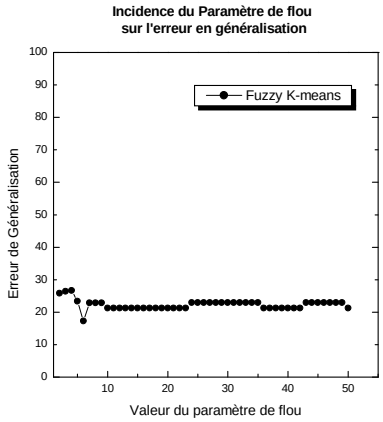

FIG. 5 – Incidence du paramètre flou sur la généralisation.

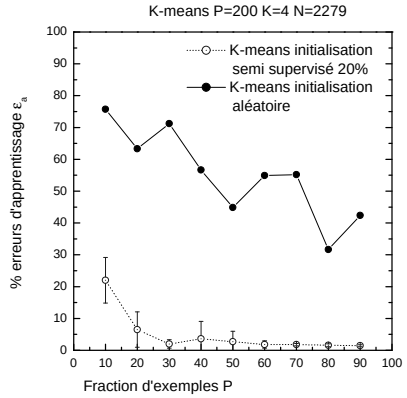

FIG. 6 – Comparaison entre une initialisation aléatoire et une initialisation semi-supervisée

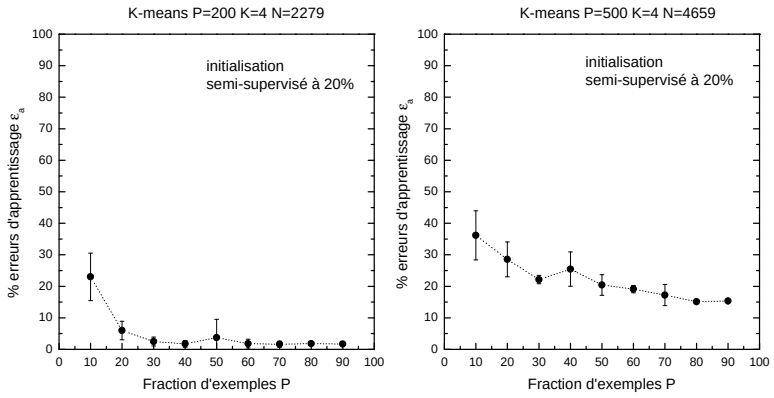

FIG. 7 – k-means $P = 200$ courriels à gauche et $P = 500$ courriels à droite.

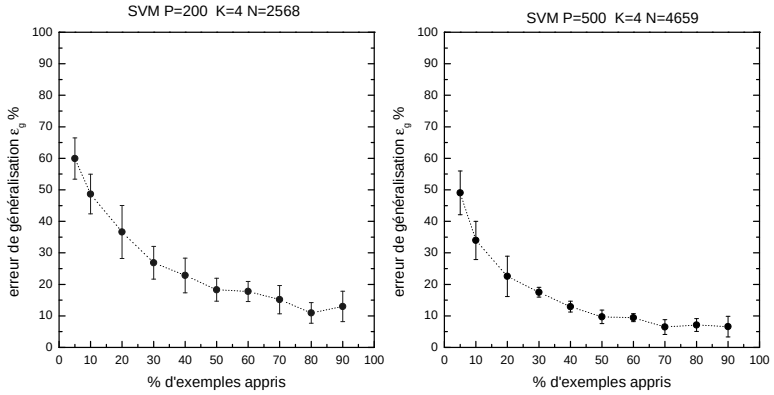

FIG. 8 – SVM $P = 200$ courriels à gauche et $P = 500$ courriels à droite.

se basant sur la classification préalable obtenue par k-means. Les corpus utilisés sont ceux de $P = 200$ et $P = 500$ courriels avec 4 classes différentes ainsi qu'un nouveau corpus de $P = 1000$ courriels avec 4 classes différentes. Les figures 9 et

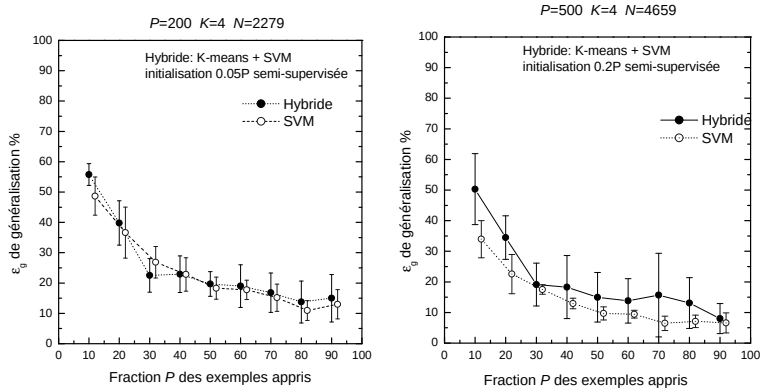

FIG. 9 – Méthode hybride, $P = 200$ courriels à gauche et $P = 500$ courriels à droite.

10 montrent que la méthode hybride obtient de très bons résultats. La performance ne se déteriore pas en augmentant la taille du corpus comme le montre la figure 10 où nous avons un corpus de $P = 1000$ courriels. Nous constatons que la courbe de la méthode hybride est très proche de celle des SVM. L'étape suivante consistera donc a améliorer les résultats obtenus avec les SVM afin de pouvoir augmenter les performances de la méthode hybride.

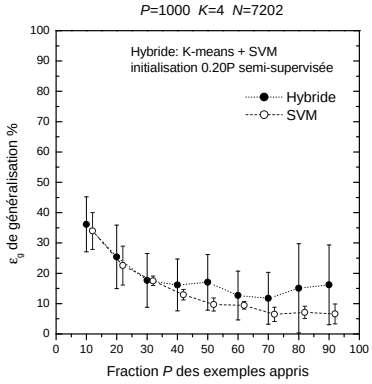

FIG. 10 – Méthode hybride, $P = 1000$ courriels.

5 Conclusion et perspectives

La tâche de classification de courriels est assez difficile en raison des particularités de cette forme de communication. Nous avons effectué dans un premier temps une comparaison entre les méthodes d'apprentissage automatique sur un corpus de référence réaliste. Cette comparaison nous a permis d'établir et de vérifier les points faibles des méthodes. Nous avons trouvé que l'apprentissage supervisé permet de mieux classer les courriels mais il demande une classification préalable qui n'est pas toujours facile à mettre en œuvre.

Par contre, bien que les résultats de k-means et de fuzzy k-means avec une initialisation aléatoire semblent mitigés, nous avons bien augmenté ces performances en apprentissage avec une initialisation semi-supervisée avec un faible nombre d'exemples. De plus, cette méthode n'a pas besoin de données étiquetées. Dans les deux cas la réduction de dimension apporte une très légère augmentation des performances et une diminution significative du temps de traitement.

Le travail effectué nous a permis par ailleurs de constater l'importance du pré-traitement, car celui-ci détermine l'espace dimensionnel pour la suite. Nous travaillons actuellement sur d'autres heuristiques de pré-traitement afin d'améliorer le filtrage et la lemmatisation telle que la suppression des accents dans les dictionnaires, les fautes par ajout de lettres, etc...

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 premiers résultats intéressants. Nous nous sommes principalement intéressé à l'amélioration de l'apprentissage non supervisé, celui-ci ayant les résultats les plus bas au départ. Les derniers résultats montrant que le système hybride est proche des résultats des SVM, la prochaine étape consistera à améliorer ceux-ci afin de pouvoir améliorer les performances globales du système. Ainsi, une implémentation des machines à support vectoriel selon l'algorithme DDAG [3] permettrait d'éliminer les régions inclas-

sifiables et donc d'améliorer les résultats de l'apprentissage supervisé. De même, l'utilisation d'une combinaison de classifieurs [7] pourrait permettre d'améliorer nos résultats. Nous prévoyons par la suite d'augmenter la taille de nos corpus de travail ainsi que le nombre de classes.

Remerciements

Ces travaux ont été réalisé dans l'équipe de Traitement Automatique du Langage Naturel Ecrit du Laboratoire d'Informatique d'Avignon. Nous tenons particulièrement à remercier Patrice Bellot, Teva Merlin, Grégoire Moreau de Montcheuil ainsi que tous les membres du LIA, dont l'aide précieuse m'a beaucoup apporté.

Références

[1] Beauregard S. "Génération de texte dans le cadre d'un système de réponse automatique à des courriels." Mémoire de maîtrise Université de Montréal, Québec, Canada, 2001.

[2] Bezdek, J.C., "Pattern Recognition with Fuzzy Objective Function Algorithms" Plenum Press, New York, 1981.

[3] Boonserm, Nitiwut "Multiclass Support Vector Machines Using Adaptive Directed Acyclic Graph" http ://bioinfo.cpgei.cefetpr.br/anais/WCCI02/IJCNN02/PDFFiles/Papers/1263.pdf

[4] Collobert R. & Bengio S. "On the Convergence of SVMTorch, an algorithm for Large-Scale Regression problem" (http ://www.ai.mit.edu/projects/jmlr/papers/volume1/collobert01a/collobert01a.pdf) 2000.

[5] deGruijter, J.J., McBratney, A.B., "A modified fuzzy k means for predictive classification"
Bock H.H.Classification and Related Methods of Data Analysis. pp. 97-104. Elsevier Science, Amsterdam.

[6] Flemm, "Un analyseur flexionnel du français à base de règles" Fiammetta Namer, TAL vol. 41 No. 2/2000 pp 523-247.

[7] Grilheres, B., Brunessaux S. and Leray P., "Combining classifiers for harmful document filtering" RIAO 2004, pp. 173-185.

[8] Kosseim L. et Lapalme G., "Critères de sélection d'une approche pour le suivi automatique du courriel" Actes de la 8ème conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN-2001), pp. 357-371, juillet 2001, Tours, France.

[9] Manning D.C. & Schütze H., "Foundantions of Statistical Natural Language Processing. The MIT Press, 2000" The MIT Press, 2000.

[10] Torres-Moreno J.M., Bougrain L. and Alexandre F.,”Database Classification by Hybrid Method combining Supervised and Unsupervised Learnings” 2003 ICANN/ICONIP 2003, pp 37-40.

[11] Torres-Moreno, J.M., Velázquez-Morales, P. and Meunier, J.G.,”Condensés de textes par des méthodes numériques” JADT 2002, V2 :723-734, A. Morin & P. Sébillot éd, IRISA, France 2002.

[12] Torres-Moreno, J.M., Velázquez-Morales, P. and Meunier, J.G., ”Cortex : un algorithme pour la condensation automatique des textes” ARCo 2001, La cognition entre individu et société ARCo 2001. Hermès Science France. pp 365 + vol 2. ISC-Lyon, pp 65-5, Décembre 2001.

[13] Torres-Moreno, J.M., Velázquez-Morales, P. and Meunier, J.G., ”Classphères : un réseau incrémental pour l’apprentissage non supervisé appliqué à la classification de textes” JADT 2000, pp 365-372, M. Rajman & J.-C. Chappelier éditeurs, EPFL, 2000.

[14] Vapnik V., ”The Nature of statistical Learning Theory (second ed.)” Springer, 1995.

[15] Vinot R., Grabar N., Valette M., ”Application d’algorithmes de classification automatique pour la détection des contenus racistes sur l’Internet” (http ://www.stud.enst.fr/ vinot/publi/ taln2003.pdf) 2003

[16] Wartel D., ”Algorithmes de clustering” (http ://www.galilei.ulb.ac.be/rd/priv_publi/ClusteringAlgorithms.pdf) 2003.