Identification de la langue et catégorisation de textes basées sur les N-grammes

Radwan Jalam [*,] — Olivier Teytaud [*,***]**

[*] *ERIC - Université Lumière Lyon 2*
Bâtiment L, 5 avenue Pierre-Mendès-France - France
4, rue Léonard De Vinci - BP 6759
69676 BRON Cedex – FRANCE

[**] *jalam@eric.univ-lyon2.fr,* [***] *oteytaud@univ-lyon2.fr*

RÉSUMÉ. *Nous rappelons tout d'abord diverses techniques utilisées en catégorisation de textes. Nous nous penchons ensuite sur une approche prometteuse: les N-grammes. Généralisable à d'autres types de données (ADN en particulier), cette méthode s'avère efficace. Nous développons un nouveau noyau pour le text mining, basé sur la dissimilarité du χ^2, et l'appliquons aux Support Vector Machines, et à un algorithme simple de type Fonctions à Bases Radiales. Le cas de la discrimination de la langue est plus détaillé en raison des spécificités des approches spécialisées sur ces problèmes, mais l'approche est illustrée sur un problème de catégorisation plus difficile et est généralisable à d'autres problèmes, notamment tous les problèmes pouvant se ramener à la classification de distributions (par exemple la classification d'images, travail en cours).*

ABSTRACT. *We recall a few techniques used in text categorization. We then study more carefully a promising approach: N-grams. Generalizable to other kinds of data sets (e.g. DNA), this approachs are efficient for practical applications. We then study a new kernel, based upon the χ^2 dissimilarity and use it with Support Vector Machines, and to a simple Radial-Basis-Function algorithm. The case of langage recognition is more detailed because of specificities of this task, but most approachs can be generalized to other problems and we make experiments on other cases; indeed our algorithm could be used for any classification of distributions (for example image classification, work in progress).*

MOTS-CLÉS : *Catégorisation de textes, Identification de langue, Distribution, N-grammes, χ^2, Support Vector Machines, Fonctions à bases radiales, Text Mining*

KEYWORDS: *Text Categorization, Language identification, Distribution, N-grams, χ^2, Support Vector Machines, Radial-Basis-Function, Text Mining*

1. Introduction & état de l'art

Plusieurs éléments représentatifs d'une langue peuvent être considérés pour de la catégorisation de langue: la présence de certains caractères [MUS 65] [SOU 94]; la présence de certains mots [SOU 94] [GIG 98]; la fréquence de n-grammes [BEE 88] [CAV 94] [DUN 94] [GRE 95]. Nous pouvons distinguer trois familles d'approches : les approches **linguistiques**, les approches **statistiques** et **probabilistes**, et les approches basées sur des méthodes générales de **classification** (réseaux de neurones, SVM...). Après avoir détaillé ces approches, nous proposerons un nouvel algorithme et verrons quels algorithmes s'avèrent efficaces dans différents cas.

1.1. *Les approches basées sur des connaissances linguistiques*

Ces approches d'identification supposent la construction des ressources linguistiques et nécessitent une connaissance préalable. Elles sont peu généralisables de la classification de langues à la catégorisation de textes.

Une approche consiste à trouver, empiriquement ou en utilisant des compétences linguistiques, des **chaînes de caractères uniques** pour chaque langue. Mais ces chaînes ne sont pas un indicateur fiable [SOU 94]; comme l'explique [DUN 94], il est difficile d'accepter d'attribuer la langue italienne à tout texte qui parle de Zucchini ou Pinocchio ou d'associer la langue française à un texte anglais qui cite le mot français «milieux». De plus, ces chaînes de caractères uniques sont relativement rares et par conséquent il est difficile de les avoir dans des passages de textes courts.

Si on définit la langue d'un énoncé par la langue des mots qui le composent, le simple recours à des **lexiques de mots** de chaque langue est suffisant pour identifier la langue : il suffit de reconnaître la langue du segment comme étant celle pour laquelle le lexique contient tous les mots. L'incomplétude des lexiques, la technicité des termes, le risque de faute (de frappes ou de reconnaissance automatique), perturbent ces méthodes [GIG 98].

Une autre approche, plus intuitive, consiste à choisir les **mots grammaticaux** comme discriminants pour l'identification de la langue [GRE 95]. L'utilisation de ces mot grammaticaux (**prépositions**, **conjonctions**, **déterminants**, **pronoms**, **adverbes**, **auxiliaires**) se justifie par le fait qu'ils permettent un diagnostic sûr; ils représentent en moyenne 50% des mots d'une phrase dans la plupart des langues et leur présence est indispensable [GIG 98]. Les mots grammaticaux ont la propriété d'être courts et en nombre limité ce qui permet d'envisager la construction de listes exhaustives. Cette approche, bien que très rapide et relativement efficace (plus que la méthode précédente), est faible sur des séquences dépourvues de tels mots, comme souvent les titres de sections [GRE 95]. En outre, étant peu informateurs, ces mots sont souvent extraits par des prétraitements lors d'indexation pour recherches à grande échelle («stop words», [SAH 99]). La méthode souffre aussi d'une difficile adaptabilité au traitement de séquences d'autres types (biologie), et de langues difficiles à segmenter en mots.

On peut combiner ces techniques et leur adjoindre la vérification de l'**alphabet** et des **affixes** (**préfixes** et **suffixes**).

1.2. *Les approches statistiques et probabilistes*

Ces approches utilisent des connaissances construites automatiquement à partir d'un corpus textuel représentatif de la langue. L'objectif est de capturer au moyen de modèles statistiques ou probabilistes certaines régularités des langues et de leur associer une fréquence ou probabilité d'apparition. Elles se généralisent de la reconnaissance de langue à la classification de textes et à part dans le cas de mots, à la catégorisation de séquences sur un alphabet discret.

Une solution pour transformer des textes en vecteurs de fréquences, plus faciles à manier pour les algorithmes d'apprentissage usuels, consiste à découper les textes en mots, quitte à ne garder que les mots les plus fréquents ou les plus discriminants. La plupart des méthodes évoquées ci-dessous pourraient se faire sur ce principe, mais nous choisissons d'utiliser une autre façon de segmenter les textes: les *n*-grammes, parfaitement généralisables à toutes formes de séquences sur un alphabet discret; les mots étant, eux, difficiles à définir pour le Chinois ou pour des séquences ADN.

Un ***n*-gramme** est une séquence de *n* caractères. Bien que, dans la littérature, ce terme désigne parfois des séquences qui ne sont pas ordonnées, ici un *n*-gramme désignera une chaine de *n* caractères consécutifs [CAV 94].

Pour un document quelconque, l'ensemble des *n*-grammes qu'on peut générer est la famille de photos qu'on obtient en déplaçant une fenêtre de *n* cases sur le corps de texte. Ce déplacement se fait par étapes, une étape (qui correspond à un caractère) pour chaque déplacement. À chaque étape une prise de photo se fait, l'ensemble des « photos » qu'on obtient constitue l'ensemble de tous les *n*-grammes qu'on puisse générer. Par exemple, les 5-grammes de la phrase «Je suis un génie», sont: je␣su, e␣sui, ␣suis, suis␣, uis␣u, . . . , etc [MIL 99]. Un **profil *n*-grammes** d'un document consiste en la liste des contiguïtés de *n* caractères les plus fréquentes, par ordre décroissant de leur fréquence d'apparition dans le document et munis de leurs fréquences. Les profils s'obtiennent en temps quasi-linéaire grâce à des tables de hachage (sous l'hypothèse que les N-grams sont convenablement répartis par la table de hachage).

La notion de *n*-grammes est introduite par Shannon [SHA 48] où il s'intéressait à la prédiction d'apparition de certains caractères en fonction des autres caractères. Depuis, les *n*-grammes sont utilisés dans plusieurs domaines comme l'identification de la parole, la recherche documentaire, etc.

Cette technique est tolérante aux déformations causées lors de l'utilisation des OCR et tolérante aux fautes d'orthographes. [MIL 99] montre qu'il y a des systèmes de recherches documentaires basés sur les *n*-grammes qui ont gardé leurs performances avec des taux de déformations de 30%, un taux avec lequel aucun système basé sur les mots ne peut fonctionner correctement.

Les systèmes d'identification de langues basés sur les *n*-grammes suivent en général le même schéma. Dans une première phase, la phase d'acquisition automatique des connaissances linguistiques, on choisit un corpus représentatif pour chaque langue puis on génère un profil caractéristique qui fera office de référence: calculer les nombres d'occurrences des différents *n*-grammes (souvent *n* est compris entre 1 et 5) et les transformer en fréquences. Dans la deuxième phase, nommée la phase de diagnostic, pour chaque texte à identifier, on construit son profil *n*-grammes et on détermine sa classe.

1.2.1. *Les méthodes de plus proche(s) voisin(s)*

Plusieurs algorithmes, utilisés pour la catégorisation de texte, sont basés sur la distance ou plus généralement sur la similarité ou dissimilarité. L'idée générale est de chercher le texte, parmi l'ensemble d'apprentissage, qui soit le plus proche du texte T à classer pour ensuite lui attribuer la classe de ce texte. La généralisation à k plus proches voisins est immédiate; il suffit de choisir la classe la plus représentée parmi les classes de k textes les plus proches. La difficulté pour ces approches est de définir une distance. En pratique, on utilise des pseudo-distances, ne vérifiant pas tous les axiomes des distances au sens mathématique.

• La **distance de Beesley**

La méthode de [BEE 88] est basée sur des modèles mathématiques linguistiques qui étaient proposées essentiellement pour décoder les textes cryptés. L'identification comporte deux phases : dans la phase d'apprentissage, établir, pour chaque langue, un profil bi-grammes (sans pourtant utiliser une lettre plus d'une fois - il ne s'agit pas du profil au sens usuel: "ordinateur" donne "or", "di", "na", etc, mais pas "rd","in") qui fera office de référence et évaluer les probabilités d'apparitions de chaque bi-gramme. Dans la phase de diagnostic: rechercher, pour le texte T à identifier, le profil de référence le plus proche en utilisant pour mesure de similarité avec la langue L le produit des fréquences des bi-grammes de T dans la langue L (méthode Bayésienne Naïve).

Cette méthode suppose la segmentation du texte en mots ce qui empêche son utilisation pour des langues dont les frontières entre les mots ne sont pas claires. Elle se limite également à prendre en compte uniquement les bi-grammes alors qu'il est important de travailler aussi avec des tri-grammes ou quadri-grammes pour mieux concerver la spécificité de chaque langue: par exemple, la séquence «tion» est typique du français et de l'anglais, si on la découpe en «ti» et «on» alors le système aura du mal à pénaliser les autres langues comme l'espagnol et le portugais qui utilisent «ti» et «on» mais pas «tion». Cela explique que la notion de profil gardée par la suite soit remaniée.

• La **distance de Cavnar et Trenkle** (CT)

Cette distance date de [CAV 94]. L'identification comporte deux phases : pendant la phase d'acquisition, il faut établir pour chaque langue un profil tri-grammes caractéristique acquis automatiquement qui fera office de référence. Ensuite il faudra rechercher, pour chaque énoncé particulier, le profil de référence le plus proche en utilisant une mesure de similarité. Cette mesure entre deux profils P_1 et P_2 se calcule comme suit:

$$CT(P_1,P_2) = \sum_{w \in P_1,\, R_{P_1}(w) < NMAX} min(|R_{P_2}(w) - R_{p_1}(w)|,DMAX)$$

avec $|x|$ la valeur absolue de x et $R_P(w)$, avec w un n-gramme et P un N-profil, désigne le rang de w dans le profil P, si w fait parti de P, et $DMAX$ dans le cas contraire (e.g., $NMAX = 500$ et $DMAX = 1000$). On conçoit facilement que cette notion un peu tranchée est moins sensible aux nuances que certaine des distances qui suivent.

• La **distance de Kullbach-Leibler**(KL)

[SIB 96] proposent une méthode qui utilise l'entropie relative de Kullbach et Leibler comme mesure de distance. Une entropie relative entre deux distributions de probabilité reflète la quantité d'informations supplémentaires nécessaire pour coder la deuxième distribution en utilisant un code optimal généré pour la première. Formellement,

$$KL(T_1,T_2) = \sum_{N_g} f_2(N_g).\log(\frac{f_2(N_g)}{f_1(N_g)})$$

ici, la somme est prise sur tous les n-grammes présents dans T_2, avec T_1 et T_2 des textes, $f_i(N_g)$ est la fréquence du n-gramme N_g dans le texte T_i. Si le n-gramme N_g est absent dans T_i, une demi-fréquence est alors ajoutée pour éviter que le score tombe vers moins l'infini.

• La **distance du** χ^2

Cette distance peut être présentée comme suit :

$$\chi^2(T_1,T_2) = \sum_{N_g} \frac{(f_1(N_g) - f_2(N_g))^2}{f_2(N_g)}$$

Dans nos expérimentations, on symétrise cette distance en utilisant $\chi^2(T_1,T_2) = 2\frac{(f_1(N_g)-f_2(N_g))^2}{f_1(N_g)+f_2(N_g)}$. Si $f_1(N_g)+f_2(N_g)$ est égal à 0, alors on remplace $\frac{(f_1(N_g)-f_2(N_g))^2}{f_1(N_g)+f_2(N_g)}$ par 0 (son prolongement par continuité).

• La **méthode de [COW 98]**

Cette méthode est intéressante pour sa capacité à mêler différents profils N-grammes (pour différentes valeurs de N) d'un même texte.

La phase d'apprentissage: [COW 98] remplit un ensemble, nommé CNP, avec des n-grammes communs ($n = 1,2,...,k,...,N$) choisis. Plus précisément, pour choisir la liste des 1-grammes qui feront partie du CNP, un poids $W(a_1)$, pour chaque

1-gramme a_1 présent dans l'ensemble d'apprentissage, est calculé de la manière suivante :$W(a_1) = -p(a_1).\log p(a_1)$. Ensuite, les 1-grammes seront classés par ordre décroissant de leurs poids. Seulement les premiers m_1 seront classés dans le CNP.

[COW 98] refait la même procédure pour les 2-grammes, k-grammes, $\ldots$, n-grammes. Afin de calculer le poids de chaque k-gramme, ils distinguent deux cas:

1. Si le $[k-1]$-gramme $a_2a_3...a_k$ n'est pas présent dans le CNP alors
$$W(a_1a_2a_3...a_k) = -p(a_1a_2a_3...a_k).\log p(a_k|a_1a_2a_3...a_{k-1})$$
2. Sinon $W(a_1a_2a_3...a_k)$
$$= -p(a_1a_2a_3...a_k).(\log p(a_k|a_1a_2a_3...a_{k-1}) - \log p(a_k|a_2a_3...a_{k-1}))$$

puis, ils classent les k-grammes par ordre décroissant de poids et placent les premiers m_k dans le CNP. Une fois l'ensemble CNP rempli, [COW 98] construit un vecteur primaire de poids PRW pour chaque k-gramme, le i-ième composant PRW_i est le poids de ce k-gramme dans la langue i. Ce poids PRW_i , pour un 1-gramme a_1 et la langue i, est calculé de la manière suivante: $PRW_i = -\log p_i(a_1)$. Pour les autres k-grammes $a_1a_2...a_k$, ce poids est: $PRW_i = -\log p_i(a_k|a_1a_2...a_{k-1})$. Si un k-gramme est absent dans une langue, alors [COW 98] définit une valeur maximale MAX, et on pose $PRW_i = MAX$. A la fin du processus, ils calculent un vecteur de poids de reconnaissance RW_n pour chaque n-gramme comme suit:

$$RW(a_1a_2a_3...a_n) = \left\{ \begin{array}{c|c} PRW(a_1a_2...a_n) & a_1a_2...a_n \in CNP \\ PRW(a_2a_3...a_n) & a_2a_3...a_n \in CNP \\ PRW(a_3a_4...a_n) & a_3a_4...a_n \in CNP \\ PRW(a_n) & a_n \in CNP \\ vecteur\ de\ MAX & autrement \end{array} \right.$$

Egalement, pour chaque langue i, [COW 98] calcule le poids moyen et sa variance. Ils segmentent l'ensemble d'apprentissage de chaque langue i en K segments $x_1x_2...x_{500}$ de taille 500 octets, puis calculent la moyenne WA_i et la variance D_i:

$$\text{pour } k \in [1,K] d_{i,k} = \sum_{s=N}^{500} \frac{RW_i(x_{s-N+1}x_{s-N+2}...x_s)}{500}$$

$$WA_i = \frac{1}{K}\sum_{k=1}^{K} d_{i,k} \qquad D_i = \frac{1}{K}\sum_{k=1}^{K}(d_{i,k} - WA_i)^2$$

La phase de diagnostic: Cette phase comporte deux étapes: la classification et la validation. Pour chaque segment de texte $X = x_1x_2x_3...x_s$ dont la classe est inconnue, on calcule son vecteur de poids $RRW(X)$:

$$RRW_i(X) = \frac{1}{S}\sum_{s=N}^{S} RW(x_{s-N+1}x_{s-N+2}...x_s)$$

On détermine alors $i^* = \arg\min RRW_i(X)$. Le texte X est de langue i^* si on a $RRW_{i^*} - WA_{i^*} \leq 1.5D_{i^*}$ sinon la langue du texte sera considérée comme inconnue. La méthode se généralise bien sûr à la catégorisation de séquences.

1.2.2. *Modèles de Markov*

Un modèle de Markov est principalement un ensemble fini d'états entre lesquels sont définies des probabilités de transition. Une distribution de probabilité sur les états est donnée aussi, qui détermine l'état initial. Formellement, soit $E = [1,n]$, et soit M une matrice $n \times n$ de nombres réels positifs tels que la somme sur chaque ligne soit égale à 1. Soit p un vecteur de n réels positifs et de somme 1. Soit X_0 une variable aléatoire égale à i avec probabilité p_i. Définissons X_{m+1} comme une variable aléatoire, égale à i avec probabilité $M_{X_m,i}$. La suite des X_i est un processus de Markov.

Etant donné un modèle de Markov A, on peut évaluer facilement la probabilité de générer un texte S avec ce modèle; avec $[z]$ le rang d'un k-gramme z parmi les k-grammes en ordre lexicographique,

$$P(S|A) = p([s_0,...,s_{k-1}])\Pi_{w \in S, |w|=k+1} M_{[w_1,...,w_{k-1}],w_k}$$

Pour simplifier cette équation, comme fait dans [DUN 94] on ajoute toujours en tête de texte un k-gramme initial, de façon à ce que $p([s_0,...,s_{k-1}]) = 1$; ainsi ce terme peut être supprimé. Nous ferons l'hypothèse que la probabilité *a priori* de chaque classe est égale aux autres, égale à $\mathcal{P}$ (quoique l'on puisse facilement tenir compte d'un a priori). Une chaîne S étant donné, $P(S)$ est égal à 1. Donc

$$P(A|S)P(S) = P(A,S) = P(S|A)P(A) \text{ se simplifie en } P(A|S) = P(S|A)\mathcal{P}$$

Pour trouver A qui maximise $P(A|S)$ parmi différents A possibles, on a juste à calculer tous les $P(S|A)$ et à trouver A qui le maximise. Ainsi, les problèmes majeurs sont les évaluations des paramètres des modèles de Markov d'une classe donnée. On pourrait simplement évaluer $M_{[w_1,...,w_{k-1}],w_k} = \frac{T(w_1,...,w_k)}{T(w_1,...,w_{k-1})}$, avec $T(w)$ le nombre d'occurences de w dans le(s) texte(s) de la langue à modéliser. Le problème est le risque que certains mots n'apparaissent jamais (par finitude de l'échantillon d'apprentissage), ceci entrainant de mauvais résultats sur des textes comportant des séquences pathologiques, quand bien même ces textes seraient complètement cohérents avec le modèle considéré par ailleurs. [DUN 94] détaille la correction de Laplace des échantillons finis, qui consiste à choisir:

$$M_{[w_1,...,w_{k-1}],w_k} = \frac{T(w_1,...,w_k) + 1}{T(w_1,...,w_{k-1}) + n} \qquad (n \text{ taille de l'alphabet})$$

D'autres approches plus compliquées sont possibles, mais selon [DUN 94] elles n'améliorent pas significativement les performances.

1.3. *Les méthodes basées sur des outils classiques de classification dans* $\mathbb{R}^n$

L'utilisation de la notion de mots ou de N-grammes permet aisément de remplacer un texte par une distribution (à chaque N-gramme on associe sa fréquence). On remplace ainsi un problème de catégorisation de chaînes par un problème de catégorisation dans $[0,1]^N$, pour lequel de nombreuses approches existent. La difficulté est la très grande dimension. L'utilisation d'**arbres de décision** nécéssite notamment une forte sélection de variables, $C4.5$ par exemple fonctionnant seulement pour des petites dimensions; or certains auteurs considèrent (voir [JOA 98]) que toutes les variables doivent être gardées. La méthode par arbres de décision bénéficie toutefois de l'avantage de la clarté de l'algorithme de classification obtenu. Les réseaux de neurones à rétropropagation, munis d'entrées par listes et non par vecteurs, peuvent par contre traiter toutes les entrées. Les **Support Vector Machines** et l'approche **Radial-Basis-Function** (RBF) ayant, pour les noyaux usuels, une complexité linéaire en la dimension et des besoins en mémoire dépendant du nombre d'exemples plutôt que de la dimension, s'avèrent efficaces pour ce problème [JOA 98].

2. Notre approche

L'encodage dans $\mathbb{R}^n$ (par N-grammes ou mots) permet l'utilisation de nombreux algorithmes, en particulier des Support Vector Machines (SVMs, voir une description dans [VAP 95] par exemple). Mais on peut utiliser les SVMs d'une autre façon: on définit $K(T_1,T_2) = exp(-d(T_1,T_2))$ avec d une mesure de dissimilarité. Nous avons choisi la distance du χ^2, sans parvenir à prouver que ce noyau vérifie la condition de Mercer (i.e. nous ne savons pas si $int_u \int_v K(u,v)g(u)g(v)dudv$ est > 0 pour tout g L^2 non nulle).

De même, on peut aussi utiliser un réseau à base de fonctions radiales (RBF) avec un encodage des textes dans $\mathbb{R}^n$ et la distance euclidienne; mais nous avons estimé plus naturel d'utiliser une distance adaptée à des distributions, par exemple la distance du χ^2. Cette méthode (du moins l'une de ses variantes) est résumée ci-dessous (on trouvera plus d'informations dans [BIS 95]):
• Soit les (T_i) la famille de textes classifiés utilisés pour l'apprentissage et les T'_i les textes à classifier.
• Soit O la matrice telle que $O_{i,j} = 1$ si T_i appartient à la classe j, -1 sinon.
• Soit $K_{i,j} = exp(-d(T_i,T_J))$
• Soit $K'_{i,j} = exp(-d(T'_i,T_j))$
• Soit W telle que $K1 \times W = O$. $K1$ est la matrice K, plus une colonne de 1 à sa droite.
• Soit $O' = K'1 \times W$.
• On assigne à T'_i la classe $argmax_k\ O'(i,k)$.

2.1. *Expérimentations: reconnaissance d'écrivain*

Nous utilisons 130 livres en français, écrits par 28 auteurs connus: Balzac, Bloy, Corneille, Diderot, Engels, Flaubert, Fourier, France, Gaberel, Gautier, Gobineau, Hugo, Huysmans, Lamartine, Leibnitz, Maistre, Maupassant, Moliere, Pascal, Racine, Renard, Rostand, Rousseau, Sand, Stendhal, Verne, Voltaire, Zola. Certains sont traduits d'autres langues; ceci et le fait que les textes sont diversement formatés (sans corrélation avec l'auteur) sont considérés comme des difficultés supplémentaires pour

Algorithme	Taux de succès
RBF avec noyau $(\chi^2)^p$ pour $p = \frac{1}{2}, \frac{1}{4}, \cdots \frac{1}{32}$ RBF avec noyau χ^2 pour $p = 1$	87.69 % 86.15 %
Multiclasse svm avec noyau χ^2	86.15 %
Multiclasse svm linéaire	78.462 %
SVM avec noyau χ^2 , $C = 1$	72.3077 %
1-PPV avec dissimilarité χ^2	70.77 %
SVM linéaire	67.69 %
1-PPV avec dissimilarité de Cavnar and Trenkle	65.38 %
1-PPV avec dissimilarité cosinus	61.54 %
1- avec dissimilarité Kullbach-Leibler	52.31 %

Tableau 1. *Résultats obtenus par Leave-One-Out sur le problème des écrivains.*

l'algorithme; le corpus d'apprentissage représente environ 30 Mo, les longueurs varient de nouvelles de quelques pages à des livres entiers. La plupart des textes proviennent du site http://cedric.cnam.fr/ABU/. Les autres proviennent de la Bibliotheque Nationale de France, http://www.bnf.fr/. Les résultats expérimentaux obtenus avec des 3-grammes sont donnés en table 1. Le taux d'erreur est évalué par Leave-One-Out. Tous ces tests sont réalisés en Octave (voir http://www.che.wisc.edu/octave/ pour une description de ce clone gratuit de Matlab). Tous les codes sources peuvent être demandés par mail aux auteurs.

On appelle "SVM multiclasse" une SVM spécialisée dans le multiclasse, comme définie dans [GUE 00]. On peut signaler que dans ce cas précis (très haute dimension, 28 classes) cette SVM est significativement meilleure que la méthode usuelle combinant des SVMs un-contre-tous comme suggéré dans [VAP 95]. On a à la fois SVM multiclasse avec χ^2 significativement meilleure que SVM avec χ^2 et SVM multiclasse linéaire significativement meilleure que la SVM linéaire. Nos essais avec modèles de Markov, par la méthode donnée par [DUN 94], ont donné de très mauvais résultats, laissant supposer que cette méthode est trop sensible à la présence de séquences pathologiques pour traiter des chaînes longues.

[YAN 99] conclut en gros que les SVM sont meilleures que les k-plus proches voisins eux-mêmes meilleurs que LLSF (voir [YAN 99]), que les réseaux de neurones multicouches basés sur la rétropropagation, eux mêmes supérieur à l'algorithme Bayésien Naïf (voir [GOO 65]). Nos essais confirment ces résultats et nous pouvons préciser, avec $>>$ signalant une différence significative avec risque d'au plus 5 %, $>$ une différence significative avec risque d'au plus 10 %, $\geq$ une différence avec risque d'au plus 15 %:

$$\{ \text{ RBF - SVM Mc } (\chi^2) \} >> \text{SVM Mc} \geq \text{SVM } \chi^2 \text{ - SVM - 1-PPV}$$

[JOA 98] explique (partiellement) le bon comportement des SVMs sur la catégorisation de texte par sa capacité à traiter de nombreuses dimensions sans avoir à sélectionner des variables. On peut noter que les RBFs bénéficient de la même caractéristique.

2.2. *Expérimentations: reconnaissance de langue*

Après ce premier benchmark, on pourrait conclure (trop vite) que les RBFs avec noyau χ^2 semblent idéales pour la catégorisation de textes. En fait la SVM multiclasse s'est avérée aussi puissante, mais les RBF sont beaucoup plus simples à implémenter et beaucoup plus rapides. Dans nos expérimentations suivantes, nous nous concentrons sur deux algorithmes: les RBFs, pour leurs bonnes performances signalées ci-dessus, et les 1-plus proches voisins, pour leur simplicité d'utilisation et leur fréquente utilisation dans ce domaine. La suite des essais a été réalisée avec des implémentations Java, basées sur la package Jama (voir http://math.nist.gov/javanumerics). Les codes sources peuvent être demandés par mail aux auteurs. La tâche consiste a reconnaître la langue d'un texte. Nous travaillons sur 5 langages: le français, l'arabe, l'anglais, l'espagnol et l'allemand. La tâche étant facile, nous la compliquons en utilisant des chaînes très courtes pour tester les algorithmes. Des tests préliminaires ont été faits pour comparer les différentes distances utilisées pour des algorithmes de plus proches voisins; nous avons obtenu les meilleurs résultats avec Kullbach-Leibler, suivis de Cavnar & Trenkle, et enfin le χ^2. La différence était forte dans le cas d'échantillons courts à reconnaître, aussi pour l'utilisation par SVM et RBF nous avons conservé le χ^2 néanmoins, en raison de sa formulation intuitivement plus susceptible de vérifier la condition de Mercer (toujours non démontrée). Nous n'avons pas effectué d'expérimentations avec la méthode de [COW 98].

Nous travaillons maintenant sur 250 échantillons de 100 bytes comme ensemble d'apprentissage, pour étudier l'influence de la segmentation pour les RBFs ou les plus proches voisins; les résultats sont donnés en table 2.

"m-regroupés profils" signifie que les échantillons d'apprentissage ont été regroupés m par m; "regroupés", que tous les textes d'une même classe sont regroupés. Le paramètre m gère ainsi un compromis entre la qualité des données (plus m est grand, plus les fichiers utilisés en apprentissage sont grands, et donc plus leurs fréquences sont bien définies) et leur quantité (plus m est petit, et plus on a de points disponibles pour la phase d'apprentissage).

On note que la dissimilarité de Kullbach-Leibler semble meilleure que celle du χ^2, notamment pour des petits échantillons en test, mais supporte mal les petits échantillons en apprentissage (comme on s'y attend en examinant le comportement de cette dissimilarité, mal régularisée, pour de petites fréquences). Ceci pourrait toutefois être amélioré en utilisant la divergence de Jensen-Shannon.

L'hyperparamètre σ^2 pour l'apprentissage RBF était facilement choisi dans le benchmark précédent (classification des auteurs), car le taux de succès était stable sur une grande plage de valeurs de σ; mais dans le cas présent (cas de très courtes

Algorithme	Hyperparamètres	TS (100)	TS (50)	TS (20)
RBF	$\sigma^2 = 10$	99.2 %	84.8 %	31.52 %
	$\sigma^2 = 100$	98 %	93.2 %	71 %
RBF (2-regroupés prof.)	$\sigma^2 = 100$	97.2 %	88 %	68.56 %
RBF (5-regroupés prof.)	$\sigma^2 = 1000$	98.8 %	94 %	80.88 %
RBF (10-regroupés prof.)	$\sigma^2 = 1000$	99.2 %	95.2 %	76.72 %
RBF (25-regroupés prof.)	$\sigma^2 = 1000$	98.8 %	94.8 %	82.4 %
RBF (regroupés prof.)	$\sigma^2 = 100$	88.4 %	80.6 %	
	$\sigma^2 = 100000$	87.6%	77.4 %	61.36 %
1-PPV	χ^2	99.2 %	96.6 %	88.4 %
1-PPV	KL			47.2 %
1-PPV (2-regroupés prof.)	χ^2	99.6 %	96.8 %	88.8 %
1-PPV (5-regroupés prof.)	χ^2	100 %	97.6 %	90 %
1-PPV (10-regroupés prof.)	χ^2	99.2 %	97.2 %	88.56 %
1-PPV (10-regroupés prof.)	KL			89.84 %
1-PPV (25-regroupés prof.)	χ^2	100 %	96.8 %	87.2 %
1-PPV (regroupés prof.)	χ^2	100 %	93 %	84.56 %
1-PPV (regroupés prof.)	KL	99.7 %	97.4 %	89.4 %

Tableau 2. *Résultats obtenus en reconnaissance des langues.*

chaines), l'efficacité variait beaucoup selon σ, ainsi qu'en fonction du paramètre de regroupement; d'où deux hyperparamètres difficiles.

3. Conclusion

Sur des données pour lesquels les fréquences sont bien évaluées, on peut finalement résumer nos résultats et ceux de [YAN 99] et [JOA 98] par:

RBF $>$ SVM Mc χ^2 $>$ SVM Mc $>$ SVM χ^2 $>$ SVM $>$ 1-PPV $>$ LLSF, C4.5, NNets $>$ NB

Avec SVM Mc la SVM multiclasse précédemment définie, SVM la SVM linéaire, LLSF comme décrit en [YAN 99], NNets des réseaux de neurones autres que SVM et NB l'algorithme bayésien naïf décrit en [GOO 65]. Notons que RBF $>$ SVM Mc n'est pas significatif en terme de performance; nous le conservons simplement en raison de l'avantage de vitesse et de simplicité.

Dans le cas de fréquences moins bien définies, les plus proches voisins semblent meilleurs que les RBF, en particulier avec de très petits échantillons d'apprentissage. Les résultats de [DUN 94] avec des modèles de Markov, avec deux langages au lieu de 5 ici, suggèrent que les modèles de Markov entraînés avec $50Ko$ de données ont environ les mêmes performances que les 1-plus proches voisins avec $25Ko$. Le taux d'erreur en cas de réponse aléatoire étant de 20% dans le cas de 5 langages et de 50% dans le cas de 2 langages, et en rappelant que les deux langues utilisées par Dunning sont testées sur le même échantillon, nous supposons les k-plus proches voisins plus adaptés pour ce problème.

Journées Francophones d'Extraction et de Gestion des connaissances.

4. Bibliographie

[BEE 88] BEESLEY K., « Language Identifier: A Computer Program for Automatic Natural Language Identification on On-Line Text », *Proceedings of the 29th Annual Conference of the American Translators Association*, 1988, p. 47–54.

[BIS 95] BISHOP C., *Neural Networks for pattern recognition*, Oxford, 1995.

[CAV 94] CAVNAR W., TRENKL J., « N-Gram Based Text Categorization », *Symposium on Document Analysis and Information Retrieval*, Las Vegas, 1994.

[COW 98] COWIE J., LUDOVIK E., ZACHARSKI R., « An Autonomous, Web-based, Multilingual Corpus Collection Tool », *Proceeding of Natural Language Processing and Industrial Applications*, Moncton, Canada, 1998, p. 142–148.

[DUN 94] DUNNING T., « Statistical Identification of Languages », rapport n° MCCS 94-273, 1994, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico.

[GIG 98] GIGUET E., « Méthode pour l'analyse automatique de structures formelles sur documents multilingues », PhD thesis, Université de Caen, France, 1998.

[GOO 65] GOOD I., *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press, 1965.

[GRE 95] GREFENSTETTE G., « Comparing Two Language Identification Schemes », *Proceedings of the 3rd International Conference on the Statistical Analysis of Textual Data (JADT'95)*, Rome, Italy, 1995.

[GUE 00] GUERMEUR Y., ELISSEEFF A., PAUGAM-MOISY H., « A new multiclass SVM based on a uniform convergence result », 2000, accepted at IJCNN'2000.

[JOA 98] JOACHIMS T., « Text Categorization with Support Vector Machines: Learning with Many Relevant Features », *Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, 1998, p. 137–142.

[MIL 99] MILLER E., SHEN D., LIU J., C.NICHOLAS, « Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System », *Journal of Digital Information*, vol. 1, n° 5, 1999.

[MUS 65] MUSTONEN S., « Multiple Disriminant Analysis in Linguistic Problems », *Statistical Methods in Linguistics*, vol. Page visitée le 15 juin 2000 à l'adresse http://www.nodali.sics.se/bibliotek/kval/smil, 1965.

[SAH 99] SAHAMI M., « Using Machine Learning to Improve Information Access », PhD thesis, Computer Science Department, Stanford University, 1999.

[SHA 48] SHANNON C., « The Mathematical Theory of Communication », *Bell System Technical Journal*, vol. 27, 1948, p. 379–423 and 623–656.

[SIB 96] SIBUN P., REYNAR J., « Language Identification: Examining the Issues », *Symposium on Document Analysis and Information Retrieval*, Las Vegas, 1996, p. 125–135.

[SOU 94] SOUTER C., CHURCHER G., HAYES J., HUGHES J., JOHNSON S., « Natural Language Identification Using Corpus-Based Models », *Hermes Journal of Linguistics*, vol. 13, 1994, p. 183–203.

[VAP 95] VAPNIK V., *The Nature of Statistical Learning*, Springer, 1995.

[YAN 99] YANG Y., LIU X., « A Re-Examination of Text Categorizarion Methods », *SIGIR'99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 1999.