# Adéquation des modèles de représentation aux méthodes de catégorisation

Simon Jaillet*, Maguelonne Teisseire*
Gérard Dray**

*LIRMM-CNRS - ISIM-Université Montpellier 2
161 rue Ada, 34392 Montpellier Cedex 5 France
{jaillet, teisseire}@lirmm.fr
**LGI2P, EMA Site EERIE,
Parc Scientifique G. Besse, 30319 Nimes France
gerard.dray@ema.fr

**Résumé.** Cet article s'interesse à la problématique de la catégorisation de documents et plus particulièrement à l'impact de la méthode de représentation des documents dans le processus de catégorisation. A partir de différents jeux de documents représentés dans un espace vectoriel tout d'abord basé sur les concepts puis basé sur une approche de type $TF\text{-}IDF$, nous évaluons les méthodes de catégorisation SVM et Rocchio. Nous comparons ensuite les deux méthodes précédentes avec une méthode de clustering flou. Nous dressons ensuite le bilan des différentes représentations des textes en terme de qualité des résultats de classification.

## 1 Introduction

Les documents numériques disponibles sont en nombre perpétuellement croissant. L'intérêt de disposer de méthodes, de techniques efficaces de classification n'est plus à démontrer et de nombreux travaux de recherche [Sebastiani, 2002, Yang et Liu, 1999] se focalisent sur cet aspect. Les résultats obtenus sont utiles aussi bien pour la recherche d'information que pour l'extraction de connaissance. L'objectif est de classer de façon automatique les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering. De façon très globale, le processus de catégorisation de document peut être décomposé selon : (1) une étape de formalisation textuelle des documents, (2) une étape d'apprentissage.

Dans cet article, nous nous intéresserons plus particulièrement à l'impact de la représentation des documents textuels (et des catégories) sur les différents algorithmes d'apprentissage (supervisés ou non). Il existe de nombreuses représentations textuelles cependant la plus utilisée est la représentation statistique de type $TF\text{-}IDF$ où chaque dimension de l'espace vectoriel correspond à un élement textuel, nommé terme d'indexation. Dans [Jaillet *et al.*, 2003], les documents sont représentés non plus en fonction des mots qu'ils contiennent mais en fonction d'une projection de ces derniers sur un ensemble fini de concepts. L'objectif est d'intégrer plus de sémantique dans la modélisation des documents. Mais enrichir les données manipulées ne permet pas

toujours d'améliorer les résultats des approches de catégorisation. Dans quelle mesure leurs performances sont-elles liées à un modèle de représentation des documents ? L'objectif de cet article est donc à partir d'une base commune de représentation des documents de comparer les résultats des différentes approches de catégorisation de références dans le domaine dont Rocchio [Rocchio, 1971] et les Support Vector Machines (SVM) [Joachims, 1998b]. Pour cette étude, la base commune est constituée de la représentation statistique de type $TF\text{-}IDF$ et de la représentation conceptuelle des documents définie dans [Jaillet *et al.*, 2003]. L'objectif des expérimentations est de mettre ainsi en évidence l'impact du modèle de représentation sur les différentes méthodes de catégorisation. L'analyse s'est également portée sur l'utilisation du modèle de représentation conceptuel (le plus riche sémantiquement) avec une approche de type clustering. Le clustering consiste à diviser les données en groupes sans connaître a priori leurs classes d'appartenance. L'ensemble des groupes obtenus n'a de sens, dans notre contexte, que par l'interprétation qui consiste à étiqueter chaque cluster par une catégorie [1]. Il devient alors possible de comparer les résultats obtenus avec ceux des méthodes précédentes Rocchio et SVM.

L'article est organisé de la façon suivante. Dans la section 2, nous présentons la problématique de la catégorisation et définissons le modèle de catégorisation textuel (MCT) utilisé. La section 3 détaille les deux modèles de représentation : le premier statistique de type $TF\text{-}IDF$ et le second basé sur les concepts. La section 4 présente les catégoriseurs : Rocchio [Rocchio, 1971], Support Vector Machine (SVM) [Joachims, 1998b] ainsi qu'une approche de clustering flou (le subtractive clustering) et leur formalisation dans le cadre du MCT proposé. Dans la section 5, nous analysons les différentes expérimentations réalisées sur des données issues de dépêches de presse. Et enfin, nous concluons en analysant l'impact de la représentation des documents dans le processus général de catégorisation.

## 2 Catégorisation de documents

La plupart des algorithmes de catégorisation se basent sur des méthodes d'apprentissage qui, à partir d'un jeu d'entraînement, permettent de catégoriser de nouveaux documents. Ce type de méthodes sont dites inductives car elles induisent de la connaissance à partir des données en entrée (les documents) et des sorties (leurs catégories).

Pour réaliser un processus de catégorisation, la première étape consiste donc à formaliser les textes afin qu'ils soient utilisables aussi bien pour les algorithmes d'apprentissage, que lors de l'étape de catégorisation. Cette étape est bien entendu cruciale car c'est elle qui permettra ou non aux méthodes d'apprentissage de produire une bonne généralisation à partir du jeu d'entraînement.

Le but de la recherche sur la catégorisation automatique de textes est donc de trouver un algorithme permettant d'assigner un texte à une ou plusieurs catégories avec le plus grand taux de réussite possible.

[1] La stratégie de la catégorie majoritaire dans le cluster est la plus souvent utilisée mais d'autres méthodes sont envisageables.

Formellement, un processus de catégorisation se définit comme une fonction :

$$\breve{\Phi} : D \times C \rightarrow \{Vrai, Faux\}$$

Avec $D$ l'ensemble des documents et $C$ l'ensemble des catégories.

L'objectif d'un processus de catégorisation est donc d'approximer la fonction précédente par une fonction $\Phi$ dans le but de maximiser une fonction d'évaluation. Le problème de la catégorisation peut donc se résumer à trouver un modèle mathématique capable de représenter, afin de comparer la "sémantique" des textes et des catégories.

Dans la suite de l'article, nous utilisons le modèle de catégorisation textuelle défini dans [Jaillet *et al.*, 2003] pour représenter les différentes étapes du processus de catégorisation qui sont : (1) L'étape de formalisation des documents, (2) L'étape de formalisation des catégories, (3) La définition d'une mesure de similarité entre documents et catégories, (4) La politique de catégorisation.

Le modèle de catégorisation textuel général $MCT_{Gen}$ se définit par le tuple : $MCT_{Gen} = (MT, MC)$. $MT$ correspond au modèle de représentation textuelle dont l'objectif est de formaliser au mieux la "sémantique" des documents au sein d'une représentation mathématique. $MT$ se définit par le tuple :

$MT(V_T, R_T, rep_T, V_C, C)$ avec,

$$\begin{cases} V_T & \text{un vocabulaire qui est un ensemble fini de dimension } \mid V_T \mid \\ T & \text{un ensemble de segments textuels tel que : } \forall t \in T, t \in V_T^* \\ R_T & \text{une représentation mathématique (espace métrique, ensemble ordonné, etc...)} \\ rep_T : T \rightarrow R_T & \text{une fonction permettant de générer une représentation } r \in R_T \text{ à partir d'un segment textuel } t \in T \\ V_C & \text{un ensemble de segments textuels fini de dimension } \mid V_C \mid \\ C & \text{un ensemble de catégories tel que : } C \subseteq \mathcal{P}(V_C) \end{cases}$$

$MC$ représente les phases d'apprentissage et de catégorisation. Le modèle de catégorisation $MC$ se définit par le tuple :

$MC = (R_C, rep_C, sim_T C, PC)$ avec,

$$\begin{cases} R_C & \text{une représentation mathématique (espace métrique, ensemble ordonné, etc...)} \\ rep_C : C \rightarrow R_C & \text{une fonction permettant de générer une représentation } r \in R_C \text{ à partir d'une catégorie c } \in C \\ sim_{TC} : R_T \times R_C \rightarrow \mathbb{R}^+ & \text{une relation entre } r_t \in R_T, r_c \in R_C \\ PC : T \times C \rightarrow \{0, 1\} & \text{une politique de catégorisation avec } t \in T, c \in C \end{cases}$$

Nous définissons aussi $\{T_{App}, T_{Test}\}$ une partition de T définissant respectivement le jeu d'apprentissage et le jeu de test. $T_{App}$ est utilisé pour construire $rep_C$, $T_{Test}$ sert seulement lors de l'évaluation.

L'interêt de ce modèle est de formaliser et de différencier chacune des étapes du processus de catégorisation : la formalisation des textes et des catégories [2] ainsi que la définition d'une mesure de similitude et d'une politique de catégorisation. Dans une problématique de catégorisation, l'intérêt de $rep_T$ (formalisation des textes) réside dans sa capacité à pouvoir "extraire" l'information du texte nécessaire à une bonne catégorisation. Quant à l'intérêt de $rep_C$ (formalisation des catégories), il réside dans sa capacité à pouvoir modéliser la notion de catégorie, c'est-à-dire extraire d'un ensemble de textes l'information qui leur est commune.

Dans la section 3 décrivant la représentation des documents, nous définissons la partie $MT$ du $MCT$ et dans la section 4 présentant les méthodes de catégorisation, nous développons la partie $MC$ du $MCT$.

# 3 Représentation des documents

## 3.1 Les différentes approches

La représentation textuelle la plus utilisée est issue de Salton dont l'implémentation la plus connue est SMART [Salton, 1971, Salton et McGill, 1983]. Dans ce formalisme vectoriel, chaque dimension de l'espace correspond à un élement textuel, nommé terme d'indexation, préalablement extrait du jeu d'apprentissage. La construction du vecteur d'un texte est déterminée par des propriétés statistiques de chacun des termes d'indexation du texte en question. C'est généralement une représentation de type $TF\text{-}IDF$ (Term Frequency times Inverse Document Frequency) qui est utilisé pour construire le vecteur d'un document.

$TF\text{-}IDF$ n'est pas le seul schéma de pondération vectoriel, mais c'est un des plus utilisés et des plus efficaces [Sebastiani, 2002]. Cependant d'autres représentations textuelles ont été expérimentées. La première concerne l'utilisation de phrases, à la place de "mots uniques", comme terme d'indexation. Mais les expérimentations effectuées n'ont pas été très fructueuses. Même si les phrases semblent posséder plus d'information sémantique, leurs propriétés statistiques ne permettent pas de définir des hypothèses statistiques fiables [Lewis, 1992]. En effet, la faible probabilité d'apparition d'une phrase ne permet pas d'approximer le risque réel de manière correcte grâce au risque empirique. Cependant, les recherches dans ce domaine restent actives. D'ailleurs les travaux de Caropreso et al. [Caropreso *et al.*, 2001] montrent une amélioration sensible des résultats grâce à l'utilisation de phrases statistiques au lieu de phrases syntaxiques.

Une autre approche de représentation textuelle "plus sémantique" se base sur le Langage Universel d'Echanges (Universal Networking Language ou UNL) défini dans [H. Uchida, 1999]. UNL est un formalisme permettant de représenter la "sémantique" de chaque document par un graphe. Toute information écrite en language naturel peut être convertie en UNL puis traduite dans n'importe quelle langue cible. La représentation de type UNL définit des liens sémantiques similaires à ceux de [Woods, 1993]. En UNL,

[2] Il est très courant que l'espace de représentation permettant de formaliser les textes et les catégories soit identique ($R_T = R_C$). Dans Rocchio par exemple, les textes et les catégories sont représentés par des vecteurs appartenant au même espace.

chaque phrase d'un document est définie par un hyper graphe où les noeuds sont des concepts et les arcs orientés des relations. Parce que la comparaison de graphes n'est pas intuitive, Shah et al. décrit dans [Shah *et al.*, 2002] une méthode de représentation de ces graphes pour être exploitée dans un processus de catégorisation. Néanmoins, le formalisme utilisé par Shah est aussi un formalisme vectoriel.

Dans [Jaillet *et al.*, 2003], les auteurs proposent une nouvelle méthode de représentation des documents. Au lieu de définir un espace vectoriel dont chaque dimension représente un terme d'indexation, souvent assimilé à un stem (radical), l'ensemble des termes est projeté sur un ensemble fini de concepts extrait d'un thesaurus. L'intérêt d'une telle méthode est de réduire les effets polysémiques du vocabulaire. En effet, deux synonymes partageront un ensemble de mêmes concepts. Cette représentation permet donc une factorisation des termes par regroupement de leur champ sémantique.

Par ailleurs, d'autres approches utilisent une représentation de type conceptuelle, c'est le cas de LSI (Latent Semantic Indexing)[Deerwester *et al.*, 1990] et de WCM (Word Category Map)[Kohonen *et al.*, 2000], afin de résoudre le problème de la synonymie. Cependant l'efficacité de ces deux méthodes dépend de la distribution de probabilité des mots au sein du jeu d'entraînement. En effet, comme aucune information n'est connue a priori, le jeu d'entrainement a une influence cruciale sur la génération de l'espace de concepts de ces méthodes, et donc de la représentation textuelle.

## 3.2 Représentation statistique des documents ($TF$-$IDF$)

La majorité des approches de catégorisation sont axées sur une représentation vectorielle des textes de type $TF$-$IDF$. Cette représentation est aussi très utilisée en recherche d'information [Sebastiani, 2002]. $TF$ (term frequency) par $IDF$ (inverse document frequency) correspond à la fréquence d'un terme multipliée par l'inverse de sa fréquence en document. Pour plus d'information sur la construction d'un tel vecteur le lecteur se refèrera à l'article [Salton et Buckley, 1988].

L'étape de représentation textuelle des documents peut-être formalisée par le modèle $MT_{TF-IDF}$ suivant :

$$\begin{cases} V_T & \text{l'ensemble des termes d'indexation de dimension } \mid V_T \mid \\ T & \text{un ensemble de textes tel que : } \forall t \in T, t \in V_T* \\ R & \text{l'espace vectoriel } \mathbb{R}^{\mid V_T \mid} \\ rep_T : T \rightarrow R & \text{une fonction permettant de générer un vecteur} \\ & \vec{r} \in R \text{ à partir d'un segment textuel } t \in T \\ & \\ C & \text{un ensemble de catégories tel que : } C \subseteq \mathcal{P}(T_{App}) \end{cases}$$

La fonction $rep_T$ se base sur les deux fonctions suivantes :

$$\begin{cases} \text{STEMMER(t)} \rightarrow \text{LISTE-STEMMES} \\ \quad \text{qui produit une liste de stemmes à partir d'un texte } t \in T \\ \text{VECTEUR(LISTE-STEMMES)} \rightarrow \mathbb{R}^{\mid V_T \mid} \\ \quad \text{qui génère un vecteur de type } TF\text{-}IDF \text{ à partir d'une liste de stemmes} \end{cases}$$

## 3.3 Représentation conceptuelle des documents

Pour permettre une telle représentation des documents, il est nécessaire de pouvoir projeter n'importe quel mot dans l'espace généré à partir d'un ensemble de concepts prédéfinis. Comme espace de concepts, nous utilisons un thésaurus composé de 873 concepts hiérarchisés en 4 niveaux.

Un thésaurus, contrairement à un dictionnaire, ne donne pas d'informations relatives au sens et à l'emploi des mots. Un thésaurus permet uniquement d'explorer à partir d'un concept (ou idée) les mots qui s'y rattachent et inversement. Par exemple, le mot "mélodie", défini par les concepts 741,781 et 784 (phrase, musique et chant) du thésaurus, sera représenté par un vecteur de dimension 873 dont toutes les composantes sont nulles sauf celles associées aux concepts 741, 781 et 784 qui seront identiques (cf. figure 1).

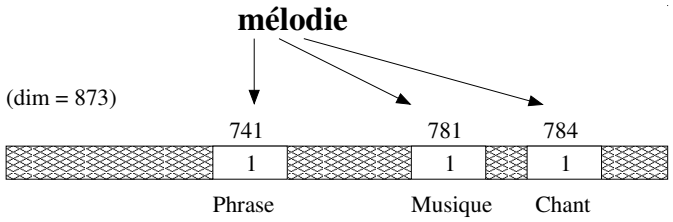


FIG. 1 – Représentation conceptuelle du mot mélodie.

Le thésaurus est donc défini comme un ensemble de couples de $L \times \mathbb{R}^{873}$ avec $L$ l'ensemble des lemmes du thésaurus.

Bien que se basant aussi sur le formalisme vectoriel pour représenter les documents, cette représentation reste fondamentalement différente de la représentation saltonnienne [Salton et McGill, 1983]. Les dimensions de l'espace vectoriel ne sont pas associées ici à des termes d'indexation mais à des concepts comme dans [Chauché, 1990].

Cependant, l'inconvénient majeur de cette représentation reste que les noms propres du document ne sont pas pris en compte. En effet, ces derniers, étant sémantiquement vides par définition, ne possèdent pas de représentation au sein du thésaurus.

Les vecteurs conceptuels des textes ont été générés à l'aide du thésaurus et du lemmatiseur défini dans [Schmid, 1994]. Même si ce type lemmatiseur reste limité pour ce qui est de l'analyse syntaxique, il offre néanmoins l'avantage de fonctionner dans toutes les langues.

L'étape de représentation textuelle des documents peut-être formalisée par le modèle $MT_{Concept}$ suivant :

$$\begin{cases} V_T & \text{l'ensemble des termes d'indexation de dimension } \mid V_T \mid \\ T & \text{un ensemble de textes tels que : } \forall t \in T, t \in V_T* \\ R & \text{l'espace vectoriel } \mathbb{R}^{\mid V_T \mid} \\ rep_T : T \rightarrow R & \text{une fonction permettant de générer un vecteur} \\ & \vec{r} \in R \text{ à partir d'un segment textuel } t \in T \\ \\ C & \text{un ensemble de catégories tel que : } C \subseteq \mathcal{P}(T_{App}) \end{cases}$$

La fonction $rep_T$, se base sur les trois fonctions suivantes :

$$\begin{cases} \text{TREE-TAGGER(t)} \rightarrow \text{LISTE-LEMMES} \\ \quad \text{qui produit une liste de lemmes à partir d'un texte } t \in T \\ \text{VECTEUR(LEMMES)} \rightarrow \mathbb{R}^{873} \\ \quad \text{qui génère un vecteur à partir d'une liste de lemmes} \\ \text{THESAURUS(l)} \rightarrow \mathbb{R}^{873} \\ \quad \text{qui associe à chaque lemme } l \in L \text{ du thésaurus un vecteur } \in \mathbb{R}^{873} \end{cases}$$

Après avoir extrait l'ensemble des lemmes d'un texte, une association, grâce à la fonction $THESAURUS$, est réalisée entre les lemmes et le vecteur qui leur est associé au sein du thésaurus. Ensuite, le vecteur conceptuel de chaque texte est calculé en fonction de la moyenne normalisée des lemmes qu'il contient :

$$\vec{r_t} = \frac{\vec{r_{l1}} + \vec{r_{l2}} + ... + \vec{r_{ln}}}{\mid\mid \vec{r_{l1}} + \vec{r_{l2}} + ... + \vec{r_{ln}} \mid\mid}$$

# 4 Les approches de catégorisation

Pour comparer l'impact de la représentation des documents sur les méthodes de catégorisation, nous avons choisi deux catégoriseurs : Rocchio [Rocchio, 1971] et les machines à vecteur de support (SVM) [Burges, 1998] ainsi qu'une approche de clustering flou, le subtractive clustering (regroupement par soustraction - RS) [Chiu, 1994]. Nous présentons ces trois méthodes selon le modèle de catégorisation MC.

## 4.1 Rocchio

Rocchio [Rocchio, 1971] est l'une des méthodes les plus anciennes en catégorisation. Nous la définissons ici dans sa version initiale. Les catégories sont représentées dans une espace vectoriel similaire aux documents. En effet, le vecteur d'une catégorie est défini comme la moyenne des vecteurs des textes qu'elle contient ($rep_C$).

Une fois les textes et les catégories représentés dans un même espace, la similitude entre un texte et une catégorie est définie par la distance euclidienne ($sim$). Par conséquent, la politique de catégorisation se résume à associer à chaque texte la catégorie dont la distance euclidienne est la plus proche ($PC$).

Sans détailler les algorithmes $rep_{C_{Rocchio}}$, $sim_{Rocchio}$ et $PC_{Rocchio}$ triviaux, nous représenterons la méthode Rocchio grâce au modèle $MC_{Rocchio}$ suivant :

$$\begin{cases} C & \text{un ensemble de catégories tel que : } C \subseteq P(T_{App}). \\ rep_{C_{Rocchio}} : C \to R & \text{la fonction permettant de générer un vecteur} \\ & r_c \in R_C \text{ à partir d'une catégorie c} \in C. \\ sim_{Rocchio} : R_T \times R_C \to \mathbb{R} & \text{la distance euclidienne entre } r_t \text{ et } r_c \text{ avec} \\ & r_t \in R_T, r_c \in R_C. \\ PC_{Rocchio} : T \times C \to \{0,1\} & \text{la politique de catégorisation définie avec } t \in T, c \in C. \end{cases}$$

## 4.2 Machine à vecteur de support

Les machines à vecteur de support (SVM) sont à l'origine de nouvelles méthodes de catégorisation [Joachims, 1998b] bien que les premières publications sur le sujet datent des années 60 [Vapnik et Chervonenkis, 1964]. Le principe des SVM consiste en une stratégie de minimisation structurelle du risque [Vapnik, 1995]. Le lecteur peut se référer à [Burges, 1998] pour une présentation générale de la méthode. En ce qui concerne son application à la problématique de catégorisation de documents, l'approche par SVM permet de définir, par apprentissage, une surface de séparation entre des exemples positifs et négatifs minimisant le risque d'erreur et maximisant la marge entre deux catégories. La figure 2 montre une telle séparation dans le cas d'une séparation linéaire par un hyperplan. Il est intéressant de remarquer qu'en réduisant le jeu d'entraînement uniquement aux vecteurs de support, l'algorithme calculerait le même hyperplan que pour le jeu d'entraînement complet. La marge se présente alors comme la plus courte distance entre un vecteur de support et "son" hyperplan.

De manière formelle, un hyperplan peut être défini par :

$$\vec{w} \cdot \vec{x} - b = 0$$

Avec $\vec{x}$ un point arbitraire, $\vec{w}$ un vecteur et $b$ le biais.

Soit $D = \{(\vec{x_i}, y_i)\}$ notre jeu d'entraînement et $y_i \in \{\pm 1\}$ définissant l'état, positif ou négatif, de l'exemple. Trouver l'hyperplan maximisant la marge séparatice $(\frac{2}{||w||})$ revient à résoudre le problème suivant :

$$\begin{cases} \text{minimiser} & \frac{1}{2}||w||^2 + C_{svm} \sum \xi_i \\ \text{sous les contraintes} & \forall i,\ y_i(\vec{w} \cdot \vec{x_i} + b) - 1 + \xi_i \geq 0 \qquad (\text{avec } \xi_i \geq 0) \end{cases}$$

$C_{svm}$ est un paramètre utilisateur. Le paramêtre $C_{svm}$ correspond à la pénalité affecté à chaque erreur. Plus $C_{svm}$ est élevé, plus l'erreur est considérée comme importante.

Grâce à une extention de cet algorithme, il est aussi possible de résoudre des problèmes qui ne sont pas linéairement séparables, mais l'amélioration obtenue pour la catégorisation de documents reste minime selon [Joachims, 1998b]. En ce qui concerne la représentation vectorielle des textes, ce sont en général les stemmes (radicaux) qui sont utilisés comme termes d'indexation.

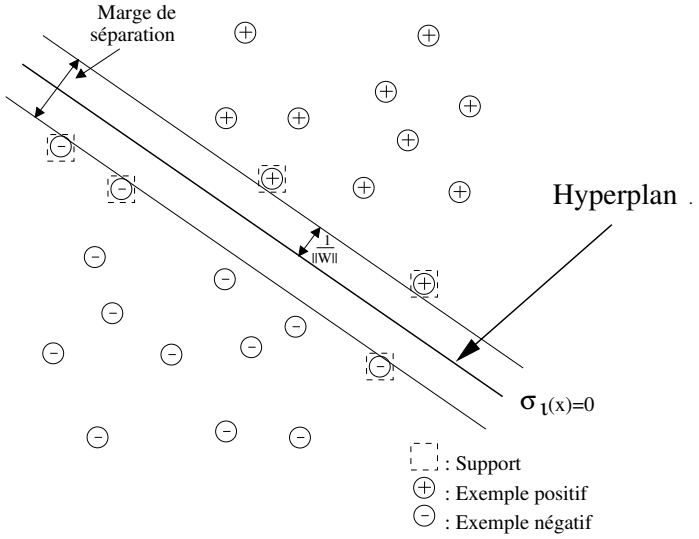


FIG. 2 – Représentation de l'hyperplan optimal

On représentera la catégorisation par SVM linéaire grâce au modèle $MC_{SVM}$ suivant :

$$\begin{cases} C & \text{un ensemble de catégories tel que : } C \subseteq P(T_{App}). \\ R_C & \text{un ensemble d'hyperplans .} \\ rep_{C_{SVM}} : C \rightarrow R_C & \text{la fonction permettant de générer un hyperplan } r_c \in R_C \text{ à partir d'une catégorie c} \in C. \\ sim_{SVM} : R_T \times R_C \rightarrow \mathbb{R} & \text{la position du point } r_t \text{ par rapport à l'hyperplan } r_c \text{ avec } r_t \in R, r_c \in R_C. \\ PC_{SVM} : T \times C \rightarrow \{0,1\} & \text{la politique de catégorisation définie avec } t \in T, c \in C. \end{cases}$$

Avec pour $rep_{C_{SVM}}$, $sim_{SVM}$ les algorithmes suivants :

**Algorithm 1:** $rep_{C_{SVM}}$

**Data** : Une catégorie $c \in C$
**Result** : Une représentation $r_c \in R_C$
**begin**
  // $\bar{c}$ est défini comme le complémentaire de $c$ sur $C$
  $\bar{c} = C - c$;
  $r_c$ =l'hyperplan maximisant la marge entre $\bar{c}$ et $c$ et minimisant l'erreur;
  return $r_c$;
**end**

Nous ne détaillerons pas $PC_{SVM}$ qui est trivialement basé sur le résultat de la fonction $sim_{SVM}$.

**Algorithm 2:** $sim_{SVM}$

**Data** : Deux représentations $\vec{r_1} \in R, r_2 = (\vec{w}, b) \in R_C$
**Result** : Un booléen $\in \{0, 1\}$
**begin**
  $\vec{w}$ = la normale de $r_2$;
  $b$ = la constante de $r_2$;
  //Calcule la position de $\vec{r_1}$ par rapport à l'hyperplan $r_2$
  **if** $(\vec{r_1} \cdot \vec{w} + b \geq 1)$ **then**
    return 1;
  **else**
    return 0;
**end**

## 4.3 *Clustering* flou

Le *clustering* flou consiste regrouper un ensemble de $n$ objets au sein de $c$ partitions différentes. L'état d'une partition est exprimé par une matrice $\mathcal{U} = (u_{ij})$ de dimension $c \times n$, où :

$$u_{ij} \in [0, 1], \quad i = 1, \ldots, c, \quad j = 1, \ldots, n \tag{1}$$

Dans ce cas, $u_{ij}$ représente "le degré d'appartenance" du $j$-ème objet à la $i$-ème partition. Les premiers travaux sur l'application du concept de sous-ensembles flous dans l'analyse du *clustering* ont été effectués par Ruspini [Ruspini, 1969].

La recherche de toutes les combinaisons possibles, à partir des données d'apprentissage, pour l'obtention de ces partitions est quasi impossible.

De ce fait, plusieurs méthodes ont été développées pour obtenir le *clustering* flou [Sato *et al.*, 1997, Bezdek, 1981, Chiu, 1994, Ruspini, 1969]. Dans cet article nous traitons de la méthode du *subtractive clustering* (Regroupement par Soustraction - RS) proposée par Chiu [Chiu, 1994] appliquée au problème de la catégorisation de textes.

On représente la catégorisation par la méthode du *subtractive clustering* par le modèle $MC_{SC}$ suivant :

$$\begin{cases} R_C & \text{Un ensemble de } \textit{clusters} \text{ et leurs fonctions d'appartenance} \\ rep_{C_{RS}} : C \rightarrow R_C & \text{L'algorithme permettant d'obtenir les } \textit{clusters} \\ & r \in R_C \text{ relatif à une catégorie c} \in C. \\ sim_{RS} : R_T \times R_C \rightarrow \mathbb{R}^+ & \text{Le degré d'appartenance du vecteur } r_t \in R \\ & \text{au } \textit{cluster } r_c \in R_C. \\ PC_{RS} : T \times C \rightarrow \{0, 1\} & \text{La politique de catégorisation avec } t \in T, c \in C. \end{cases}$$

Le principe du *subtractive clustering* est décrit par l'algorithme 3 $rep_{C_{SC}}$.
Les paramètres de départ du RS sont $R_a$, $R_b$, $\varepsilon_{inf}$ et $\varepsilon_{sup}$ où :

- $R_a = [r_{a_1}, \ldots, r_{a_p}]$ est un vecteur rayon qui définit d'une part, la taille de la zone servant à calculer la densité de voisinage, et d'autre part, la taille des groupements. Chacune des composantes $r_{a_j}$ spécifie le rayon dans la $j^{ème}$ dimension.

– $R_b = [r_{b_1}, \ldots, r_{b_p}]$ est un vecteur qui définit la distance minimale entre deux groupements dans chaque dimension.
– $\varepsilon_{inf}$ et $\varepsilon_{sup}$ sont des scalaires servant à arrêter le processus de sélection des groupements.

Le choix des rayons $r_{a_j}$ est déterminant car il influe fortement sur le résultat final du regroupement. Leurs valeurs sont fixées suivant la base de données étudiée. Quant aux autres paramètres, nous obtenons en pratique des résultats plutôt satisfaisants en posant $r_{b_j} = 1.5\, r_{a_j}$, $\varepsilon_{sup} = 0.5$ et $\varepsilon_{inf} = 0.15$. La valeur de ces paramètres est estimée empiriquement par Chiu.

**Algorithm 3:** $rep_{C_{RS}}$

*Etape 1*

1. Fixer les paramètres $R_a$, $R_b$, $\varepsilon_{inf}$ et $\varepsilon_{sup}$.
2. Pour chaque point $x_i$, initialiser la densité $P_{x_i}$ suivant l'équation

$$P_{x_i}(1) = \sum_{l=1}^{n} e^{-4 \sum_{j=1}^{p} \left( \frac{x_{i,j} - x_{l,j}}{r_{a_j}} \right)^2} \tag{2}$$

3. Accepter $x_1^*$ comme un groupement tel que $P_{x_1^*} = max_i(P_{x_i})$.

*Etape 2*

1. Mettre à jour chaque $P_{x_i}$ suivant l'équation

$$P_{x_i}(k+1) = P_{x_i}(k) - P_{x_k^*}(k)\, e^{-4 \sum_{j=1}^{p} \left( \frac{x_{i,j} - x_{k,j}^*}{r_{b_j}} \right)^2} \tag{3}$$

2. Poser $P_{x_k^*} = max_i(P_{x_i})$.
3. Si $P_{x_k^*} > \varepsilon_{sup}\, P_{x_1^*}$
   Alors accepter $x_k^*$ comme un groupement, et aller à 1.
4. Si $P_{x_k^*} < \varepsilon_{sup}\, P_{x_1^*}$ et $P_{x_k^*} \geq \varepsilon_{inf}\, P_{x_1^*}$
   Alors poser $d_{min}$ comme la plus petite des distances entre $x_k^*$ et tous centres des groupements trouvés précédemment $x_l^*$ :

$$d_{min}^2 = \min_{l=1,..,(k-1)} \left\{ \sum_{j=1}^{p} \left( \frac{x_{k,j}^* - x_{l,j}^*}{r_{a_j}} \right)^2 \right\} \tag{4}$$

   Sinon aller à 6.
5. Si $d_{min} + \frac{P_{x_k^*}}{P_{x_1^*}} \geq 1$
   Alors accepter $x_k^*$ comme un groupement, et aller à 1.
   Sinon rejeter $x_k^*$, initialiser le potentiel de $x_k^*$ à 0, et aller à 2.
6. Arrêter la procédure de sélection.

Le RS procède en deux étapes. Dans la première étape, il évalue une fonction pour le guider dans la recherche de centres de groupement. Cette fonction correspond à une mesure de densité de voisinage effectuée à l'intérieur d'un noyau gaussien. L'emploi d'un noyau gaussien permet de pondérer l'influence des points : un point proche du centre a un rôle plus important dans la mesure de la densité $P_{x_i}$ qu'un point éloigné. La largeur du noyau est définie par le vecteur rayon $R_a$.

Dans la seconde étape, le RS recherche itérativement les centres de groupements. L'algorithme sélectionne d'abord le point dont la densité est maximale. Puis, pour éviter de choisir un autre groupement trop proche, toutes les mesures $P_{x_i}$ sont mises à jour de sorte que la mesure d'un point se trouvant à proximité du point sélectionné se voit diminuer de manière importante. Cette mise à jour de la densité a peu d'influence sur la mesure d'un point qui se trouve à une distance supérieure à $\|R_b\| = \sum_{j=1}^{p} r_{b_j}^2$ du point sélectionné. Le RS réitère plusieurs fois le processus jusqu'à ce que le test d'arrêt soit vérifié.

Les points 4 à 6 de la seconde étape servent à tester l'arrêt du processus de sélection. Le test d'arrêt du RS est défini par un intervalle. Si la mesure du point sélectionné tombe dans l'intervalle, alors nous acceptons le point sélectionné comme centre de groupement suivant une certaine condition. En l'occurrence, la somme entre $d_{min}$ et le rapport $P_{x_k^*}/P_{x_1^*}$ doit être supérieure à 1. Si la condition n'est pas vérifiée, alors le point n'est pas retenu comme centre de groupement, et nous réitérons. Enfin, si la mesure du point sélectionné dépasse l'intervalle, alors l'algorithme s'arrête.

Il est important de préciser qu'à ce stade de la méthode, aucune information sur les classes d'appartenance des textes n'a été prise en compte. C'est uniquement dans la stratégie finale de catégorisation ($sim_{SC}$, $PC_{SC}$) que sera utilisée cette information.

La relation $sim_{TC}(\vec{r_1}, x_k^*)$ représente le degré d'appartenance $\mu_k(\vec{r_1})$ du vecteur $\vec{r_1}$ au centre $k$ de coordonnées $x_k^*$ et de rayon d'influence $R_a$.

La politique de catégorisation $PC_{SC}$ consiste à étiqueter chaque cluster par la classe majoritaire à l'intérieur de celui-ci. Lorsqu'un nouveau vecteur est présenté, la classe du cluster possédant le plus grand degré d'appartenance lui est attribuée.

# 5 Expérimentations

Les deux méthodes de catégorisation sont testées sur les deux types de représentation proposés : la représentation statistique (de type $TF\text{-}IDF$) et la représentation par concept. L'approche par clustering, quant'à elle, n'est appliquée qu'à la représentation conceptuelle des documents.

## 5.1 Les données

Pour la réalisation des expériences, nous avons utilisé un jeu de données composé de 8239 dépêches de presse réparties en 28 catégories (cf. tableau 1). Pour effectuer la représentation conceptuelle des textes, un espace vectoriel de dimension 873 a été construit à partir d'un Thésaurus de référence. Enfin, un ensemble de 10974 stems (radicaux) différents a été extrait du jeu d'entraînement pour la représentation de type $TF\text{-}IDF$.

**Algorithm 4:** $sim_{RS}$

**Data** : Deux représentations $\vec{r_1} \in R_T, r_2 = (\vec{x_k^*}, R_a) \in R_C$
$x_k^* = [x_{k,1}^*, ..., x_{k,p}^*]$ $R_a = [r_{a_1}, ..., r_{a_p}]$.
**Result** : Un réel $\in [0, 1]$
**begin**

La fonction d'appartenance $\mu_k$ associée au groupement est définie par :
Soit $v = [\, v_1 \,, \ldots \,, \, v_p \,] \in \mathbb{R}^p$,

$$d_k = {}^t(v - x_k^*)\, M_k\, (v - x_k^*) \text{avec} \quad M_k = \begin{bmatrix} 1/r_{a_1}^2 & & 0 \\ & \ddots & \\ 0 & & 1/r_{a_p}^2 \end{bmatrix} \tag{5}$$

et,

$$d_{k,j} = |v_j - x_{k,j}^*| \tag{6}$$

on a,

$$\mu_k(v) = e^{-4\, d_k^2} = e^{-4 \sum_{j=1}^{p} \left( \frac{d_{k,j}}{r_{a_j}} \right)^2} \tag{7}$$

où $d_k$ est la distance euclidienne pondérée entre le groupement $x_k^*$ et l'élément $v$, et $d_{k,j}$ est la distance par dimension $j$ entre le groupement $x_k^*$ et l'élément $v$.

**end**

## 5.2 Estimation des performances des modèles

### 5.2.1 Mesures des performances

Pour évaluer les performances des méthodes de catégorisation, nous utilisons la mesure $F_\beta$ [Rijsbergen, 1979] :

**Définition 1** *$F_\beta$ :*

$$F_\beta = \frac{(\beta^2 + 1)\pi_i \rho_i}{\beta^2 \pi_i + \rho_i}$$

$F_\beta$ est basée sur les notions de rappel (nombre de textes bien classés sur le nombre total de textes à catégoriser) et de précision (nombre de textes bien classés sur le nombre de textes classés dans cette catégorie) définies ci-dessous :

**Définition 2** *Rappel et précision :*

$$\pi_i = \frac{VP_i}{VP_i + FP_i} \quad , \quad \rho_i = \frac{VP_i}{VP_i + FN_i}$$

*avec $VP_i$, $FP_i$, $FN_i$ définissant, pour une catégorie $i$, respectivement les textes bien classés, les textes assignés par erreur ainsi que les textes omis par le classifieur.*

| Nom de la catégorie $i$ | Nombre de documents |
|---|---|
| Nouvelles internationales | 943 |
| Agriculture-Agroalimentaire | 305 |
| Banque-Finance-Assurance | 352 |
| Administration-Institution | 310 |
| Automobile-Construction mécanique | 278 |
| Aéronautique-Naval-Défense | 303 |
| Enseignement-Formation | 268 |
| Electronique-Construction électrique | 229 |
| Communication-Media-Pub-Culture | 353 |
| Distribution-Commerce | 256 |
| Hotellerie-Restauration | 217 |
| Immobilier-BTP-Logement | 229 |
| Industries Extractives-Matières | 306 |
| Industries Diverses | 254 |
| Internet-Commerce électronique | 219 |
| Informatique-Bureautique-SSII | 270 |
| Luxe-Mode-Textile | 259 |
| Santé-Pharma-Chimie | 311 |
| Services aux entreprises | 170 |
| Services aux particuliers | 171 |
| Télécoms | 325 |
| Transports-Logistique | 322 |
| Achat-Logistique | 263 |
| Direction | 300 |
| Finance | 331 |
| Marketing-Communication | 193 |
| Ressources humaines | 278 |
| Commercial Vente | 224 |

TAB. 1 – Nombre de documents par catégories

Par la suite, nous utiliserons la politique du "microaveraging" pour évaluer le classifieur dans sa globalité. C'est la mesure la plus utilisée pour comparer des méthodes de catégorisation entres elles [Sebastiani, 2002, Yang, 1999].

**Définition 3** *microaveraging :*

$$\pi^{\mu} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|}(VP_i + FP_i)} \quad , \quad \rho^{\mu} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|}(VP_i + FN_i)}$$

Cette mesure accorde la même importance à la performance de chaque document contrairement à la "macro-averaging" qui réalise une moyenne par catégorie.

### 5.2.2 Validation croisée

Pour éviter les phénomènes de biais introduits par le découpage de la base de données en ensembles d'apprentissage et de test, nous avons opté pour une validation croisée de nos méthodes. Nous avons séparé la base initiale en dix sous-ensembles tout en conservant la proportion des catégories. Puis nous avons appliqué les méthodes présentées précédemment sur neuf des sous-ensembles et estimer les performances sur le dixième sous-ensemble. Ces opérations ont été répétées dix fois. Nous avons ainsi obtenu dix mesures de performances pour chaque méthode de catégorisation. Mesures que nous avons ensuite agrégées par une moyenne pour obtenir la performance globale de chaque méthode.

## 5.3 Résultats

Pour la méthode des SVM, nous avons utilisé son implémentation linéaire à l'aide du package SVM$^{Light}$ [Joachims, 1998a] avec comme paramètre $C_{svm}$ défini dans $\{5, 20, 50\}$ pour la représentation conceptuelle et $C_{svm}$ défini dans $\{0.1, 10, 100\}$ pour la représentation de type $TF\text{-}IDF$. La méthode du *subtractive clustering* a été implémentée sous MATLAB. Les meilleurs résultats ont été obtenus pour des paramètres $ra$ fixés tous égaux à 0.85, les autres paramètres étant réglés en utilisant les valeurs préconisées par Chiu [Chiu, 1994]. Cette méthode a été utilisée uniquement sur la représentation conceptuelle. La dimension de l'espace de représentation $TF\text{-}IDF$ ne permet pas d'utiliser cette méthode dans sa version originale. Celle-ci doit faire l'objet de modifications permettant de définir une fonction d'appartenance adaptée à cet espace.

Le Tableau 2 présente les résultats obtenus par les trois méthodes en utilisant la représentation par vecteurs conceptuels. Dans cet espace, la méthode SVM donne les résultats les meilleurs, proches cependant la méthode de Rocchio. Les résultats de la méthode du SC originale sont, quant à eux, un peu en retrait par rapport aux résultats de ces méthodes. Une première analyse plus détaillée montre que les *clusters* identifiés par l'algorithme ne possède pas de catégorie à majorité franche. En effet, la plus part d'entre eux possèdent plusieurs catégories en proportions très proches. Ce problème semble être du à une localisation très dense des vecteurs dans une portion réduite de l'espace (l'angle le plus élevé entre deux vecteurs est d'une dizaine de degrés). Dans ces conditions, la distance euclidienne utilisée dans le calcul des fonctions d'appartenance n'est pas suffisamment discriminante pour déterminer les meilleurs *clusters* à l'aide d'une méthode non supervisée. Cependant, les résultats obtenus par cette méthode sont assez prometteurs pour que les travaux sur le *clustering* flou soient poursuivis ; notamment, pour le choix des fonctions d'appartenance et des distances à utiliser.

Le Tableau 3 résume les résultats de la validation croisée obtenus par les méthodes de Rocchio et de SVM sur la représentation $TF\text{-}IDF$. Dans cet espace, à l'inverse du précédent, c'est la méthode de Rocchio qui donne les meilleurs résultats. Cette expérience met tout d'abord en évidence les choix critiques de l'espace de représentation et de la méthode de classification dans une problématique de catégorisation automatique de documents. De plus, il montre l'influence du corpus sur les méthodes de catégorisation. En effet, si SVM reste le meilleur catégoriseur pour les corpus de type Reuters [Sebastiani, 2002, Yang et Liu, 1999], ce n'est pas le cas ici pour notre cor-

| Echantillon de la Validation Croisée | $F_1^\mu$ sur les 28 categories | | | | |
|---|---|---|---|---|---|
| | Rocc. | SVM avec $C_{svm}$ égale à | | | SC |
| | | 5 | 20 | 50 | |
| 1 | 0.34 | 0.36 | 0.39 | 0.38 | 0.27 |
| 2 | 0.38 | 0.41 | 0.42 | 0.39 | 0.27 |
| 3 | 0.37 | 0.42 | 0.46 | 0.43 | 0.32 |
| 4 | 0.36 | 0.40 | 0.47 | 0.46 | 0.29 |
| 5 | 0.36 | 0.44 | 0.48 | 0.45 | 0.29 |
| 6 | 0.39 | 0.46 | 0.51 | 0.50 | 0.29 |
| 7 | 0.37 | 0.44 | 0.48 | 0.44 | 0.29 |
| 8 | 0.34 | 0.43 | 0.46 | 0.46 | 0.32 |
| 9 | 0.33 | 0.40 | 0.46 | 0.43 | 0.30 |
| 10 | 0.35 | 0.42 | 0.46 | 0.43 | 0.31 |
| *Moyenne* $F_1^\mu$ | 0.36 | 0.42 | 0.46 | 0.44 | 0.30 |

TAB. 2 – Résultats des méthodes de catégorisation sur la représentation conceptuelle des textes

pus francophone. Cependant, la catégorisation à base de SVM reste stable pour les deux types de représentation textuelles contrairement à Rocchio qui s'écroule lors de la représentation conceptuelle des textes.

La mauvaise performance de Rocchio sur la représentation conceptuelle s'explique par le fait suivant : les dimensions les plus "valuées" des vecteurs ne sont pas forcement les plus dicriminantes. En effet les concepts discriminants sont généralement des concepts "rares" ayant une faible valuation au sein des vecteurs. [Jaillet *et al.*, 2003] ont d'ailleurs montré la "non-optimalité" du cosinus, comme mesure de similarité, sur ce type de vecteurs. En revanche, SVM se montre pratiquement stable sur les deux types de représentation. Cette stabilité s'explique par le fait que SVM utilise un algorithme d'apprentissage qui minimise le risque structurel. Par conséquent l'importance de chacune des dimensions (c'est à dire des concepts) ne dépend pas uniquement de la valuation de ce dernier.

# 6 Conclusion

Les expériences effectuées dans cet article permettent de mieux comprendre les particularités des méthodes de catégorisation. Tout d'abord, les résultats obtenus soulignent la stabilité de la méthode de catégorisation basée sur les SVM quelque soit la représentation des documents utilisée. Néanmoins, pour la représentation de type $TF$-$IDF$, SVM, considérée comme la meilleure méthode de catégorisation [Sebastiani, 2002, Yang et Liu, 1999], a obtenu des performances inférieures à celles de Rocchio sur notre jeu de dépêches francophones. Enfin, même si le modèle de représentation est plus riche, l'utilisation de méthodes de catégorisation classiques ne permet pas toujours d'améliorer les performances. C'est pourquoi, il est souvent nécessaire de proposer de

| Echantillon de la Validation Croisée | $F_1^\mu$ sur les 28 categories | | | |
|---|---|---|---|---|
| | Rocc. | SVM avec $C_{svm}$ égale à | | |
| | | 5 | 20 | 50 |
| 1 | 0.51 | 0.38 | 0.38 | 0.38 |
| 2 | 0.55 | 0.44 | 0.44 | 0.44 |
| 3 | 0.56 | 0.48 | 0.48 | 0.47 |
| 4 | 0.56 | 0.51 | 0.52 | 0.51 |
| 5 | 0.59 | 0.50 | 0.50 | 0.50 |
| 6 | 0.60 | 0.51 | 0.51 | 0.51 |
| 7 | 0.58 | 0.49 | 0.50 | 0.48 |
| 8 | 0.55 | 0.48 | 0.49 | 0.48 |
| 9 | 0.52 | 0.50 | 0.52 | 0.50 |
| 10 | 0.53 | 0.47 | 0.48 | 0.47 |
| *Moyenne* $F_1^\mu$ | 0.55 | 0.47 | 0.48 | 0.47 |

TAB. 3 – Résultats des méthodes de catégorisation sur la représentation de type $TF$-$IDF$ des textes

nouveaux classifieurs réellement adaptés comme la méthode des deux écarts définie pour une représentation des textes basée sur les concepts [Jaillet *et al.*, 2003].

Cette étude permet d'envisager plusieurs perspectives. Tout d'abord, il serait intéressant d'approfondir les expérimentations afin d'évaluer l'influence du corpus sur les méthodes de représentation. Les jeux de données du type "Nouvelles" ou d'articles plus longs vont modifier nécessairement le choix de la méthode de représentation. Dans quelle mesure est-il intéressant d'intégrer une modélisation plus sémantique des textes pour des articles courts ? Dans quelle mesure les performances du catégoriseur sont-elles améliorées et à quel prix ? Il serait également intéressant d'envisager une représentation mixte, couplant les deux méthodes de représentation, intégrant les avantages de la représentation statistique ($TF$-$IDF$) et conceptuelle donnant ainsi la possibilité d'utiliser des classifieurs standards.

# Références

[Bezdek, 1981] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York, Plenum Press, 1981.

[Burges, 1998] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

[Caropreso *et al.*, 2001] M. Caropreso, S. Matwin, et F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Text Databases and Document Management : Theory and Practice*, pages 78–102. 2001.

[Chauché, 1990] J. Chauché. Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. *TA Information*, 31/1 :17–24, 1990.

[Chiu, 1994] S. L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2 :267–278, 1994.

[Deerwester *et al.*, 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, et Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6) :391–407, 1990.

[H. Uchida, 1999] T. Della Senta H. Uchida, M. Zhu. *The UNL, A Gift for a Millennium*. UNU Institute of Advanced Studies, 1999.

[Jaillet *et al.*, 2003] Simon Jaillet, Maguelonne Teisseire, Jacques Chauche, et Violaine Prince. Classification automatique de documents : Le coefficient des deux écarts. In *INFORSID*, pages 87–102, Nancy, 2003.

[Joachims, 1998a] T. Joachims. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods : Support Vector Machines*. 1998.

[Joachims, 1998b] Thorsten Joachims. Text categorization with support vector machines : learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, 1998.

[Kohonen *et al.*, 2000] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, et A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3) :574–585, 2000.

[Lewis, 1992] D.D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *ACM SIGIR '92*, pages 37–50, 1992.

[Rijsbergen, 1979] C. J. Van Rijsbergen. *Information retrieval*. Butterworths, London, 2 edition, 1979.

[Rocchio, 1971] J. Rocchio. Revelance feedback in information retrieval. In *in the SMART Retrieval System : Experiments in Automatic Document Processing*, pages 313–323, 1971.

[Ruspini, 1969] E. H. Ruspini. A new approach to clustering. *Inform. Control.*, 15(1) :22–32, 1969.

[Salton et Buckley, 1988] G. Salton et C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5) :513–523, 1988.

[Salton et McGill, 1983] G. Salton et M. J. McGill. *Introduction to modern information retrieval*. 1983.

[Salton, 1971] G. Salton. The smart retrieval system – experiments in automatic document processing, 1971.

[Sato *et al.*, 1997] S. Sato, Y. Sato, et L. C. Jain. *Fuzzy Clustering Models and Applications*. Physica-Verlag Heidelberg, A Springer-Verlag Company, 1997.

[Schmid, 1994] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, 1994.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorisation. In *Proceedings of ACM Computing Surveys*, volume 34, pages 1–47, 2002.

[Shah *et al.*, 2002] C. Shah, B. Chowdhary, et P. Bhattacharyya. Constructing better document vectors universal networking language (unl). In *Proceedings of International Conference on Knowledge-Based Computer Systems*, 2002.

[Vapnik et Chervonenkis, 1964] V. Vapnik et A. Chervonenkis. A note on one class of perceptrons. *Automatic and Remote Control*, 25, 1964.

[Vapnik, 1995] V. Vapnik. *The Nature Of Statistical Learnig Theory*. Springer, 1995.

[Woods, 1993] W. Woods. What's in a link : Foundation for semantic network. *Journal of Documentation*, 49 :188–207, 1993.

[Yang et Liu, 1999] Y. Yang et X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, 1999.

[Yang, 1999] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2) :69–90, 1999.

RNTI - E -