Classification de texte et estimation probabiliste par Machine à Vecteurs de Support.

Anh-Phuc.Trinh[1]

[1] Laboratoire d'Informatique de Paris 6, Université de Pierre et Marie Curie,
104, avenue du Président Kennedy, 75016 Paris.
anh-phuc.trinh@poleia.lip6.fr

Résumé : La classification de documents D en classes pré-déterminées Y est simplement présentée comme le problème d'estimation probabiliste de la probabilité a posteriori $P(Y/D)$. Nous présentons ici une méthode basée sur le modèle de Machine à Vecteurs de Support afin de réaliser cette tâche. Il y a deux approches correspondant à deux niveaux de présentation des corpus : en documents et en phrases, que nous avons analysées dans ce défi.

Mots-clés : Machine à Vecteur de Support, Classification de Texte, Estimation Probabiliste

1 Introduction

La classification de texte est une des tâches primitives dans le domaine du Traitement Automatique des Langues (TAL), de la Recherche d'Information et des Algorithmes d'Apprentissage (AA). Les premiers efforts sont venus de la Recherche d'Information, avec (Salton et al., 1983), qui ont transformé leurs documents en vecteurs de termes. Le poids Tf*Idf (Salton et al., 1988) a été approuvé et utilisé comme la formule « standard » pour un traitement de documents.

De plus en plus, nous avons besoin de représenter les documents à un niveau plus profond, comme dans le cas des documents structurés ou semi-structurés et les documents XML qui ont une structure arborescente. Cependant, les chercheurs continuent de s'intéresser aux documents en texte simple car ils sont plus proches des langues naturelles, même si les documents structurés sont plus faciles à représenter sur les machines. Si les documents en plein texte peuvent être représentés à un niveau plus profond, nous estimons que la classification peut être améliorée par rapport aux documents entiers. Ainsi, allons-nous diviser les documents en phrases puis les analyser.

En matière de modèles d'apprentissage (MA) pour classifier des documents, nous pouvons choisir soit le modèle d'apprentissage non supervisé, soit le supervisé. Nous décidons de prendre le modèle d'apprentissage supervisé appelé Machine à Vecteurs de Support (Vapnik, 1998). Sur le problème d'estimation probabiliste par la MVS, les articles pertinents sont (Vapnik, 1998), (Platt, 2000), (Friedman et al., 1996), (Hastie et al., 1996), (Tax et al., 2002) et (Wu et al., 2004). Les deux premiers articles décrivent la classification binaire, les quatre derniers traitent du sujet de la classification de multi-classes. Nous avons également trouvé une thèse relative à la classification des documents en texte par la MVS (Joachim, 2002).

La tâche proposée par DEFT07 consistait à classifier les documents d'un corpus en deux classes (corpus de débats parlementaires) ou en plusieurs (les trois autres corpus). Nous avons donc décidé d'utiliser les méthodes de (Platt, 2000) et de (Wu et

al., 2004). En réalité, le dernier article poursuit le travail du premier, donc leurs formules sont identiques.

2 Présentation des corpus

Dans cette section, nous avons étudié les différences entre les corpus pour obtenir leurs propriétés statistiques. Afin de les pré-traiter, la technique de sac de mots a été appliquée, et nous donnons également au fur et à mesure les résultats de cette technique avec des commentaires.

2.1 Représentation générale des quatre corpus du défi

Il y a quatre corpus différents dans le cadre du défi : « Critiques de films », « Tests de jeux vidéo », « Relectures d'articles » et « Débats parlementaires ». Tous sont au format XML, donc ils ont besoin d'être traités correctement, tout en gardant les balises qui contiennent des informations indispensables. L'outil XML de Java a été utilisé lors du traitement de ces quatre corpus, cela nous permet d'extraire un sac de mots pour chaque corpus et de saisir des informations d'évaluation primaires de chaque document.

Les données des quatre corpus viennent de sources très variées. Les deux premiers semblent venir de deux sites Web de divertissement (vidéo, livres, spectacles, bandes dessinées, jeux vidéo). Les textes qui les constituent sont très variés, se composant souvent d'un petit résumé du contenu du jeu ou du livre, puis de commentaires personnels. A première vue, le niveau "mauvais", "moyen" ou "bon" semble facile à classifier en tenant compte du sens des phrases. Les deux autres, "débats parlementaires" et " relectures d'articles" semblent issus de sources plus officielles. D'une part, les débats parlementaires comprennent toujours des phrases courtes avec des termes politiques et très formels. Nous pouvons facilement définir l'intention du locuteur et ainsi classer son discours dans la catégorie "pour" ou " contre". D'autre part, nous remarquons que, dans le corpus de relectures d'articles, les textes sont plus riches et souvent il y a beaucoup de commentaires. Au niveau de la structure, les documents sont composés de phrases constituant un en-tête et sont structurés jusqu'à l'identifiant n° 275 . A partir de cet identifiant, les documents n'ont plus de phrase d'en-tête. Ce corpus nous a donné des résultats différents par rapport aux autres.

Parmi les corpus donnés, celui des relectures d'articles est le plus petit : 1471 Ko, le plus grand corpus est celui des débats parlementaires : 24 620 Ko. Par rapport aux autres défis que nous avons rencontrés, ces données sont assez grandes et en même temps assez variées. Le nombre de phrases de chaque document est très variable : le corpus "débat parlementaire" a souvent une phrase pour chaque avis ; le corpus "jeux vidéo" a en moyenne un paragraphe (de chaque avis) assez long par rapport aux autres documents. Le corpus le plus particulier est "relectures d'article" : il commence par des paragraphes assez structurés : originalité, commentaire ... puis finit par supprimer toute structure et n'a plus qu'une phrase courte pour chaque avis. Ce sont des données brutes et sans pré-traitement. Cela oblige les participants à choisir les techniques pour les pré-traiter. Ce fait augmente le temps de pré-traitement des données. Nous avons choisi la technique de sac de mots. C'est une technique classique et simple pour traiter les données textuelles que nous vous présentons maintenant.

Corpus d'apprentissage	Taille	Nombre de phrases	Nombre de documents
Critiques de cinéma, spectacles, livres, BD et CD	4Mb	34622	2074
Test de jeux vidéo	17Mb	145543	2537
Relectures d'articles de conférences	1,4Mb	11473	881
Débats parlementaires	24Mb	160399	17299

Tableau 1 – Les corpus d'apprentissage

2.2 Sac de mots

Nous pouvons constater que le sac de mots (William et al, 1995) est un dictionnaire de vocabulaire particulier. C'est une technique qui filtre les mots d'un document, puis ajoute les informations sur chaque mot filtré telles que : les indices de fréquence du mot dans le document ou dans les autres documents qui constituent le corpus.

Avec les indices d'un sac de mots, nous pouvons convertir un document en un vecteur, ainsi nous pouvons représenter un document dans un espace vectoriel. Après avoir constitué le vecteur qui correspond au document étudié, nous pouvons utiliser les formules mathématiques pour calculer puis définir la classification du document. D'après nos calculs, le sac de mots du corpus de "relectures d'articles" est plus grand par rapport à la taille de ce corpus, ce qui veut dire que les mots qui le constituent sont très variés et très diversifiés : un mot n'apparaît souvent qu'une seule fois dans le document et parfois n'existe pas dans les autres documents du corpus. Les trois corpus qui restent ont un sac de mots assez proportionnel à leur taille.

Comme nous avons montré ci-dessus, la technique du sac de mots est une technique très simple, qui récupère les mots d'un document sans prendre en compte le sens de ces mots. Faute d'outils logiciels qui puissent traiter les mots d'une façon plus approfondie : diviser les mots selon les thèmes, leur signification, ou leur domaine, etc ... nous avons adopté une solution statistique pour traiter le corpus : prendre uniquement en compte la fréquence d'un mot dans un document et dans le corpus. Une des limites de cette technique réside dans le fait d'être très sensible à la casse : par exemple, le sac de mots va prendre "BOn" et "bon" pour deux mots, ce qui augmente sans raison pertinente la taille du sac de mots. C'est la raison pour laquelle, en utilisant cette technique, nous avons tendance à diminuer le nombre de mots sans trop perdre d'information (voir la section 4.2). Le sac de mots est également utilisé afin de représenter une phrase du document (voir la section 3.2), nous constatons que l'apparition d'un mot est de la même importance que la fréquence de ce mot dans une phrase.

Corpus d'apprentissage	Nombre total de mots	Taille du sac de mots
Critiques de cinéma, spectacles, livres, BD et CD	792214	48489
Test de jeux vidéo	3084878	56185
Relectures d'articles de conférences	218588	13395
Débats parlementaires	3738562	46654

Tableau 2 – Les sacs de mots de chaque corpus d'apprentissage

Nous avons ci-dessous un ensemble des accents et numéraux enlevés du corpus, le reste constitue le sac de mots. Nous utilisons la classe « Tokenizer » de Java pour récupérer les instances. Les sacs de mots sont ensuite vérifiés plusieurs fois manuellement pour assurer que les mots filtrés gardent bien leur sens initial. Comme nous avons précisé dès le début, le sac de mots est une technique simple, utilisé dans le cas où nous n'avons ni dictionnaire, ni thésaurus. D'autre part, l'un des avantages de cette technique est que l'information est conservée telle quelle : si l'utilisateur utilise des mots spécifiques, le sac de mots les ramasse quand même et les considère comme les autres mots.

. \(\) *:, ?!* +-%&°$'0123456789*[]/<>;\n§=@#|{}} {}

3 Approches

Nous avons étudié deux approches sur les quatre corpus d'apprentissage. La première est de représenter le document comme l'unité à analyser, et l'autre est de représenter la phrase comme l'unité à analyser. La MVS est le modèle d'apprentissage et nous avons utilisé sa fonction d'estimation probabiliste pour rétablir la distribution a posteriori $P(Y/D)$

3.1 Représentation des documents

3.1.1. Définition et formules

$$P(Y/D) \text{ avec } D=\left(mot_1:poids_1, mot_2:poids_2, \ldots, mot_n:poids_n\right)$$

Où

Y correspond aux classes pré-déterminées

mot_i correspond à chaque entrée du sac de mots qui a été présenté en section précédente.

La probabilité a posteriori $P(Y/D)$ présente une mise en correspondance (mapping) de documents avec leur propre classe. La probabilité de chaque classe Y sachant le document D nous permet de dire à quelle classe ce document D appartient. Notre stratégie est la suivante : nous avons d'abord essayé de représenter des documents de la manière où ils sont les plus séparables possible, ensuite le modèle d'apprentissage MVS (Machine à Vecteur de Support) va les discriminer par son algorithme d'apprentissage non linéaire qui permet de séparer des données bruitées. Enfin, une estimation probabiliste de la classification qui vient d'être construite va conclure notre tâche, et cette estimation probabiliste a besoin de prendre en compte la capacité de discrimination des données par le modèle d'apprentissage MVS.

3.1.2. Le poids Tf*Idf

Le poids Tf*Idf (Salton et al., 1988) a été largement utilisé dans le domaine de la Recherche d'Information et de la Fouille de données Textuelles. Ce poids mesure statistiquement l'importance d'un mot d'un document par rapport au corpus entier. Cette importance est représentée à la fois par son nombre d'apparitions dans ce document et par l'inverse du nombre de documents contenant ce mot dans le corpus.

Il y a beaucoup de formes du poids Tf*Idf (Church et al., 1995), nous avons décidé d'employer celui qui suit

$$\text{TfxIdf} = \text{Term\_Frequency X Inverse\_Document\_Frequency}$$

avec

$$Idf = \ln\left(\frac{D}{(d_i \text{ contient } m_j)}\right)$$

Nous prenons le logarithme naturel du nombre de documents contenant ce mot, dans la mesure où la valeur de cette fonction varie entre $(0, +\infty)$. Grâce à ce choix, le poids Tf*Idf devient une valeur réelle et chaque document va se transformer en un vecteur réel $\Re^n$. Quand le poids Tf*Idf d'un mot est grand, cela nous indique que ce mot apparaît de nombreuses fois dans ce document et rarement dans le corpus entier, autrement dit, ce mot est très valable, parce qu'il permet de discriminer son document d'origine par rapport aux autres. Nous pouvons également mesurer la proximité ou l'éloignement entre deux documents par le cosinus (ou le produit scalaire normalisé) de l'angle que forment les vecteurs qui les représentent. L'apprentissage non supervisé basé sur cette mesure peut placer un document dans une classe en utilisant simplement un seuil adapté aux données. Cette approche classique n'assure pas que des données bruitées soient correctement classifiées par le modèle d'apprentissage. Le bruit des données existe toujours pour des raisons subjectives, par exemple, des mots que nous avons choisis ne sont pas représentatifs, leur fréquence dans chaque document ne suffit pas pour éloigner ce document des autres dans l'espace vectoriel.

La sélection des mots réduit la dimension de l'espace vectoriel dont chaque document est un vecteur et en même temps augmente la capacité de discrimination entre les documents. La dimension de l'espace vectoriel correspond à la complexité du modèle d'apprentissage, c'est-à-dire que plus il y a de mots à prendre en compte, plus il y a de variables à évaluer. Nous devons tenir compte d'un ensemble de mots qui sont souvent utilisés en français : des pronoms, des prépositions, des conjonctions etc... Leurs valeurs de poids sont toujours très faibles parce qu'ils apparaissent dans tous les documents, donc nous pouvons les enlever du sac de mots. Cependant cette technique ne s'applique que si la taille du corpus est assez grande et le vocabulaire du sac de mots est large. Les résultats de cette technique vont également être présentés (voir la section 4.2). Néanmoins, nous ne les avons pas soumis dans le cadre de ce défi.

Débats parlementaires		
mot	nombre total	idf
de	205135	17105
la	111334	16126
l	88199	15754
à	76529	15557
le	75384	15577
les	72674	14753
des	71309	14544
et	67158	14702
que	51388	14375
du	39091	12114

Les dix mots les plus fréquents dans le corpus de débats parlementaires

3.1.3. Estimation probabiliste par MVS (Machine à Vecteur de Support)

a) Définition et formules (Vapnik, 1998) (Muller et al., 2001)

Une Machine à Vecteur de Support est une technique de discrimination. Elle consiste à séparer deux (ou plusieurs) ensemble de points de données par un (ou plusieurs) hyperplan(s). Considérons un ensemble de points de données $\{(D_1,Y_1),(D_2,Y_2),\ldots,(D_l,Y_l)\}$ avec $Y_i=\{-1,+1\}$. Un hyperplan $w\cdot D-b=0$, qui sépare cet ensemble des points de données, possède également deux hyperplans parallèles $w\cdot D-b=+1$ et $w\cdot D-b=-1$ tels que la marge qui les sépare est égale à $2/\|w\|^2$. Le problème est de minimiser l'inverse de cette marge, c'est-à-dire,

$$\min\frac{1}{2}\|w\|^2 \text{ soumis à } Y_i\times(w\cdot D_i-b)\geq 1 \quad \forall i=\overline{1,l} \quad (3)$$

La fonction de signe $f(D)=signe(w\cdot D-b)$ nous indique de quel côté de la marge un document D se trouve, nous l'appelons une fonction de décision. En cas de données bruitées, l'hyperplan ne permet pas de séparer totalement certains points de données $\{D_i,Y_i\}_1^l$. En présentant des pénalités d'erreurs non nulles $\xi_i\geq 0$, chaque point de données D_i possède une valeur de pénalité, donc l'optimisation devient un compromis entre une grande marge et des pénalités d'erreurs.

$$\min\frac{1}{2}\|w\|^2 \text{ soumis à } Y_i\times(w\cdot D_i-b)\geq 1-\xi_i \quad \forall i=\overline{1,l} \quad (4)$$

b) Estimation probabiliste en cas de classes multiples

Le cas le plus simple est la classification binaire $Y_i=\{-1,+1\}$, correspondant au corpus de débats parlementaires, la distribution a posteriori $P(Y/D)$ est estimée par la fonction sigmoïde, c'est-à-dire :

$$P(Y=+1/D)=\frac{1}{(1+\exp(A\times f(D)+B))} \quad (5)$$

où A et B sont estimés en réduisant au minimum la fonction négative de vraisemblance. La vraisemblance de cette distribution des données positives est la fonction

$$L(z)=\sum_{i=1}^{l}\left(t_i\log(p_i)+(1-t_i)\log(1-p_i)\right) \quad (6)$$

Avec $z=(A,B)$ et $t_i=\frac{Y_i+1}{2}$ est la probabilité de la cible.

(Platt et al., 2000) nous propose deux techniques afin de bien estimer les A et B : la première en partageant le corpus d'apprentissage en deux (70% pour entraîner la MVS avec la fonction de décision $f(D)$, les 30% restants pour estimer A et B dans la fonction sigmoïde), la deuxième est d'utiliser la validation croisée en n-partitions.

Dans un cas plus compliqué, la classification en k-classes (k>2), correspondant aux autres corpus, nous avons normalement deux stratégies pour appliquer la technique de discrimination : "un-contre-les autres" ou "un-contre-un". Il s'agit de construire plusieurs hyperplans (k avec celle de "un-contre-les autres" et k(k-1)/2 avec celle de "un-contre-un"). A chaque hyperplan, nous avons une distribution proportionnelle de la classification binaire entre deux classes différentes

$$rij \approx P(Y=i/Y=i \text{ ou } j,D) \text{ avec } i \neq j \quad (7)$$

sur un sous-ensemble de documents dans le corpus. Basé sur la stratégie "un-contre-un", (Wu et al., 2004) ont proposé une méthode assez coûteuse pour reconstituer la distribution de multi-classification $P(Y/D)$ à partir de cet ensemble $R=\left|rij\right|_{i\neq j}$ en résolvant le problème d'optimisation suivant

$$\frac{1}{2}\sum_{i=1}^{k}\sum_{j\neq i}\left(r_{ji}P(Y=i/D)-r_{ij}P(Y=j/D)\right)^2 \quad (8)$$

$$\text{soumis à } \sum_{i=1}^{k}P(Y=i/D)=1, P(Y=i/D)\geq 0,\ \forall i$$

La fonction est obtenue en introduisant (7) dans (8)

$$P(Y=i/Y=i \text{ ou } j,D)\cdot P(Y=j/D)=P(Y=j/Y=i \text{ ou } j,D)\cdot P(Y=i/D)$$

Certainement, nous avons également d'autres méthodes pour estimer la probabilité a posteriori $P(Y/D)$ comme (Friedman et .al, 1996), (Price et al. 1995), (Hastie et al., 1998). Alors que les deux premières sont simples, la dernière est proche du problème d'apprentissage statistique. Les distributions proportionnelles $R=\left|rij\right|_{i\neq j}$ jouent le rôle des données d'apprentissage et les $P(y=i/D)$ sont des variables qui doivent être évaluées en minimisant la distance de Kullback-Leiber (KL).

Corpus d'apprentissage	A1	B1	A2	B2	A3	B3
Critiques de cinéma, BD, livres, spectacles et CD	-44,275	38,623	-13,75	11,289	-17,165	-14,213
Test de jeux vidéo	-13,266	9,42	-4	1,87	-20,59	-11,33
Relectures d'articles de conférences	-1,205	0,558	-1,161	-0,47	-3,062	-1,771
Débats parlementaires	-4,176	-2,689				

Tableau 3 Les paramètres A et B de la fonction sigmoïde.

Le tableau ci-dessus montre trois paires de $\left(A_i,B_i\right)\ i=1,3$ correspondant à trois hyperplans qui sont également nos trois classificateurs. Ces paramètres $\left(A_i,B_i\right)$ ont été estimés par la validation croisée avec 5-partitions. A chaque nouvelle donnée d'entrée D, par exemple une donnée de test, nous avons besoin de résoudre le problème (8) afin de trouver les informations probabilistes $P(Y=i/D)$. La complexité de (8) est proportionnelle au nombre de classes pré-déterminées $(k=3)$, néanmoins le problème (8) est un problème d'optimisation, ainsi donc nous avons eu besoin de beaucoup temps pour prédire D. Cela est peut-être un inconvénient de cette méthode, en particulier, si l'ensemble de données de test est large.

La Figure 1 illustre une représentation partielle des données de test du corpus Critique, le trait continu représente la probabilité proportionnelle entre les deux classes « bon » et « mauvais », autrement dit r_{ii}, le trait discontinu représente cette probabilité a posteriori : $P(Y=bon/Y=bon \text{ ou } mauvais,D)$. Nous pouvons observer que la variabilité de cette probabilité a posteriori est souvent plus basse que la probabilité proportionnelle après avoir résolu le problème (8).

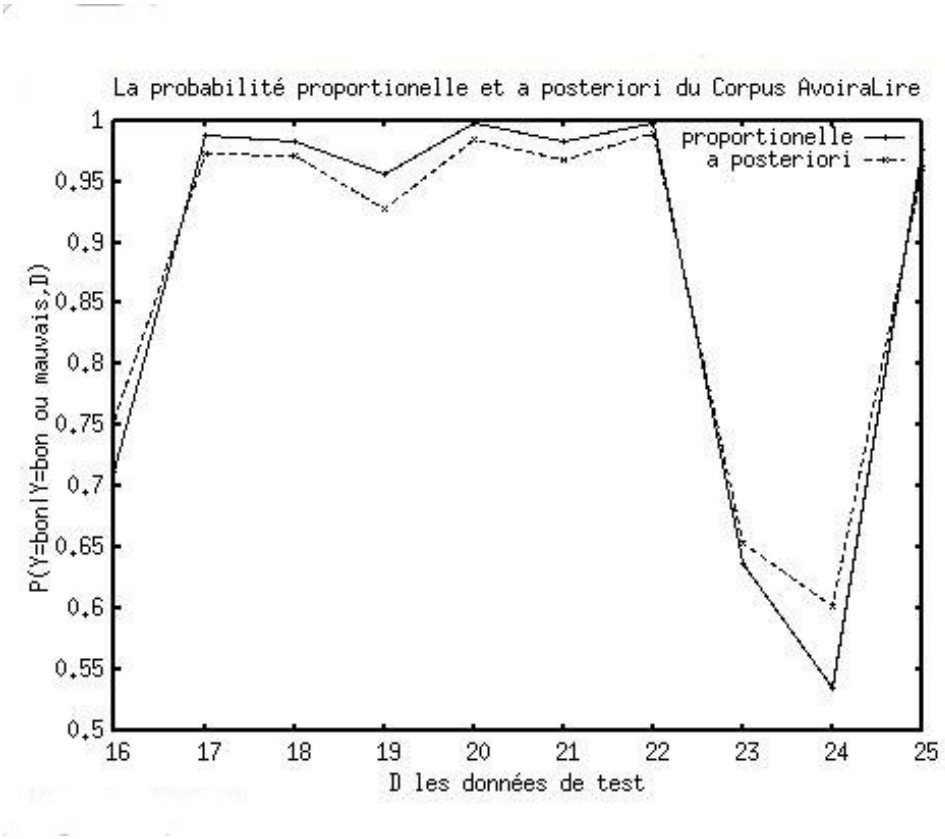

Figure 1 – Les courbes des probabilités a posteriori et proportionnelle du corpus des Critiques

3.2. Extraction locale de documents

3.2.1. Définition et formules

$$P(Y/D)=P\left(Y/phrase_1,phrase_2,\ldots,phrase_m\right)=\sum_{j=1}^{m}P\left(Y/phrase_j\right)\ avec\ \ j=\overline{1,m} \quad (9)$$

Où

$$phrase=\left(mot_1:poids_1,mot_2:poids_2,\ldots,mot_n:poids_n\right)\ avec\ poids=\left[0,1\right]$$

Jusqu'ici, nous avons considéré le document comme l'unité à analyser. Un document peut être autrement décrit comme une série de phrases, chaque phrase présentant sémantiquement une idée complète de l'auteur. Cette hypothèse nous conduit à la deuxième approche de notre travail. Comme chaque document se compose de plusieurs phrases, nous avons besoin de déterminer une technique pour réunifier les informations des phrases. Nous avons décidé simplement de prendre en charge des informations qui sont représentées par la probabilité a posteriori de chaque phrase $P\left(Y/phrase_j\right)$ dans un document. Enfin, nous procédons à une normalisation afin d'obtenir la probabilité a posteriori totale $P(Y/D)$. Le but de cette approche est de découvrir l'extraction locale de documents.

3.2.2. Représentation de la phrase par un vecteur binaire

D'abord les phrases sont extraites des documents, elles sont ensuite converties en un vecteur binaire. Nous constatons que l'apparition d'un mot est aussi importante que la fréquence de ce mot dans une phrase, ainsi elles sont représentées par les vecteurs binaires. Le vecteur binaire est souvent très faible en matière de nombre d'indices, c'est-à-dire, nous avons vu que des phrases ne contiennent qu'un seul mot. Nous espérons qu'au niveau de la phrase, les documents seront bien caractérisés et que la MVS pourra séparer facilement nos documents. En cas de documents spéciaux, par exemple, le corpus de relectures d'articles, il y a plusieurs phrases qui n'ont qu'un mot d'en-tête (originalité, commentaire...), ainsi la MVS vise à ranger toutes ces phrases dans une seule classe (voir la section 4.2).

Corpus d'apprentissage	Phrase la plus longue	Longueur moyenne de phrases	Phrase la plus courte
Critiques de cinéma, spectacles, livres, BD et CD	200	22	1
Test de jeux vidéo	171	21	1
Relectures d'articles de conférences	206	19	1
Débats parlementaires	324	23	1

Tableau 4 Les phrases pour chaque corpus d'apprentissage.

3.2.3. Discrimination locale et vraisemblance globale

La Figure 2 illustre une représentation partielle des données de test du corpus Critiques. Le trait discontinu représente la probabilité a posteriori $P(Y=mauvais/D)$ dans le cas des phrases binaires, le trait continu représente la même probabilité dans le cas de documents. Comme dans la figure précédente (voir la section 3.1.3.b), nous pouvons constater que la variabilité de la probabilité a posteriori de phrases est souvent plus basse que celle de documents : nous pouvons dire d'une autre façon que celle des documents est plus « pointue » que celle des phrases. La probabilité des documents est plus discriminante que la probabilité des phrases. Nous répétons encore une fois que le fait de distinguer les données ne signifie pas que le modèle a une bonne valeur de F-score. On arrive fréquemment à une valeur de précision plus grande en diminuant la valeur de rappel.

La MVS est un modèle de discrimination. Il en résulte que sa valeur de précision est en général plus élevée que celle du rappel. Si nous décidons de discriminer localement des documents, la valeur de précision semble encore plus élevée (voir la section 3.1.3.b et 4.2), néanmoins la valeur moyenne entre eux (F-score) est réduite. Nous estimons qu'il existe toujours un compromis entre la tendance de discrimination locale et celle de vraisemblance globale. Pour revenir sur la section d'estimation probabiliste précédente, après l'étape de discrimination locale entre sous-ensemble de données, nous avons besoin de rétablir la probabilité a posteriori totale P(Y|D) en minimisant la distance de Kullback-Leiber K(z||D).

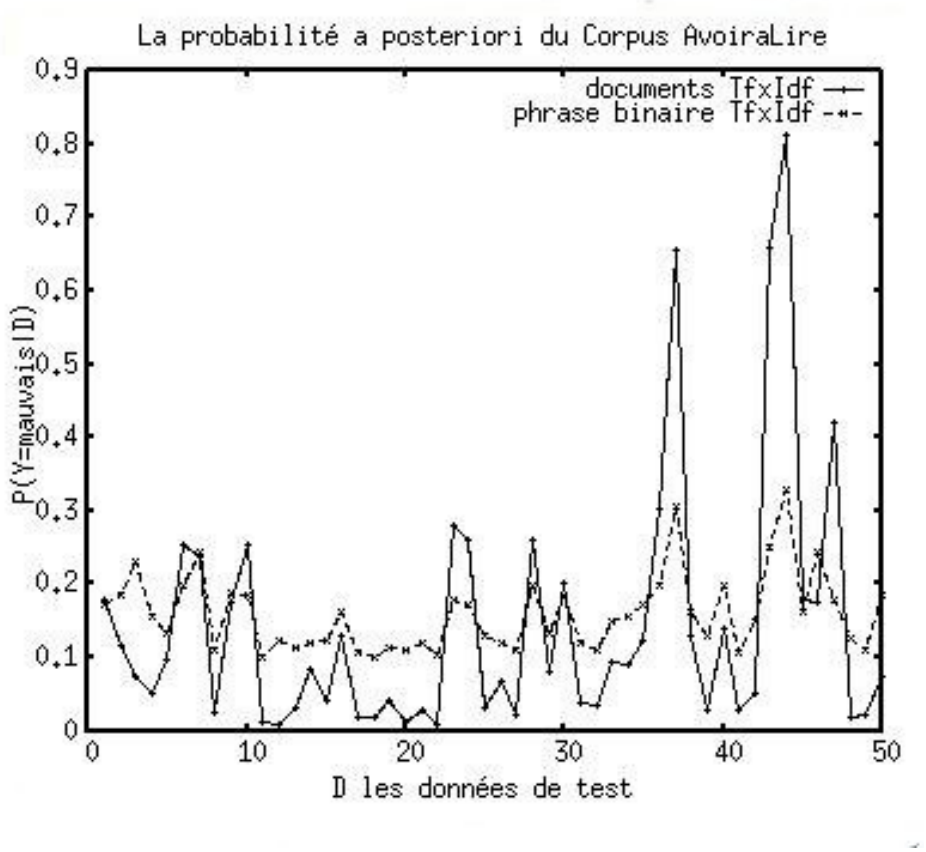

Figure 2 : La probabilité a posteriori du corpus Critiques en deux approches.

4 Résultats

Dans cette section, nous donnons les résultats définitifs des deux étapes d'apprentissage et de test. Les résultats entre les deux étapes sont assez cohérents et stables, la valeur d'exactitude (A) que nous présentons vient de (Wu et al., 2004). Les trois valeurs de précision (P), de rappel (R) et de F-score (F) sont venues de DEFT07. Les corpus d'apprentissage ont été répartis pour 60% en données d'entraînement et 40% en données de test. Et voici la liste des valeurs d'évaluation :

$$(A)\text{ccuracy} = \frac{\text{nombre correct}}{\text{nombre total}}$$

$$(P)\text{récision} = \frac{\text{nombre correct}}{\text{nombre total de cible}}$$

$$(R)\text{appel} = \frac{\text{nombre correct}}{\text{nombre total de prédict}}$$

$$(F)\text{score} = \frac{2 \times P \times R}{P + R}$$

4.1 Résultats soumis

Nous remarquons que les résultats dans les données de test et les données d'apprentissage sont assez homogènes. Premièrement, la valeur de précision (P) est souvent un peu plus élevée que la valeur de rappel (R) quand on utilise MVS. Deuxièmement, la valeur moyenne entre ces deux valeurs (F) diminue quand on considère le niveau local de la phrase. Quand on analyse la phrase, la valeur de précision augmente considérablement mais ne compense pas la perte sur le rappel.

Diviser le corpus d'apprentissage 60% et de test 40%		Les résultats de test		
Corpus d'apprentissage	(A)ccuracy	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,60576	0,553125	0,53125	0,541954
Test de jeux vidéo	0,70866	0,678847	0,640234	0,658975
Relectures d'articles de conférences	0,42937	0,458529	0,400009	0,427275
Débats parlementaires	0,70885	0,684401	0,667297	0,675741

Tableau 6 (F-score, Précision, Rappel) Les résultats de test dans le cas des documents

Regardons de plus près le cas où l'unité à analyser est un document. Les corpus des articles à lire ont des valeurs assez petites car ils contiennent peu d'information et ces informations sont répétitives. Nous mettons particulièrement l'accent sur ce corpus pour deux raisons. La première raison est qu'il est petit, les informations sont homogènes donc non condensées, ce qui conduit à la réduction des valeurs (P, R, F). Ainsi nous ne pouvons pas représenter séparément les documents. La meilleure solution est de donner des caractéristiques aux documents en les décrivant davantage. Par exemple, nous pouvons ajouter les caractéristiques concernant la longueur du document en comptant les mots de celui-ci, ou même ne pas éliminer les accents dans le sac de mots, c'est-à-dire considérer un accent comme un mot. La deuxième raison est l'utilisation de masse de ce type de document. Avec l'expansion de l'Internet, les utilisateurs ont de plus en plus tendance à remplir les formulaires préexistants qu'à écrire un long paragraphe. L'utilisateur trouve que c'est plus facile de remplir un formulaire, mais rechercher et/ou prédire l'information dans un formulaire constitue un défi. D'une autre façon, on peut considérer le corpus des articles comme un échantillon de textes, avec une petite quantité de textes appartenant à un grand ensemble de textes, et notre mission est de reconstituer ce grand ensemble.

Au départ, nous avons obtenu une valeur d'exactitude (A) très significative avec les deux corpus Débats et Jeux vidéo (les valeurs A de ces deux corpus sont homogènes et sont légèrement supérieures à 70%). Nous nous concentrons ensuite sur l'analyse des deux corpus moins volumineux qui sont celui des Relectures et celui des Critiques. Ce qui est intéressant se trouve justement dans le volume de ces deux corpus. Comme ils sont tous les deux de petite taille, le filtrage des documents prend beaucoup moins de temps par rapport aux deux premiers corpus. Ce qui reste à faire est de diviser les documents en phrases. De cette manière on augmente considérablement les données à apprendre, autrement dit, on augmente les données du défi. Un exemple simple : le corpus Relectures d'articles contient 881 documents qui sont composés de 11473 phrases (voir le tableau 1). Ainsi, le nombre de vecteurs augmente de 881 à 11473.

Diviser le corpus d'apprentissage 60% et de test 40%		Les résultats de test		
Corpus d'apprentissage	(A)ccuracy	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,55528	0,819650	0,349448	0,489993
Test de jeux vidéo	0,56102	0,788606	0,458202	0,579625
Relectures d'articles de conférences	0,45197	0,508588	0,432168	0,467274
Débats parlementaires	0,67360	0,683378	0,647842	0,665136

Tableau 7 (F-score,Précision,Rappel) Les résultat des tests dans le cas des phrases

Le tableau ci-dessus contient les résultats de la représentation au niveau local. Pour les corpus auxquels on s'intéresse particulièrement, tels que les Relectures d'articles, la remarque est que toutes les valeurs augmentent en même temps (A, P, R, F) par rapport à la représentation au niveau des documents. Ce résultat renforce la validité de notre principe de calcul des valeurs. Nous attirons l'attention sur ce fait car ces mêmes valeurs, dans les autres corpus, sont moindres. Ce n'est pas surprenant vis-à-vis des trois autres. Nous pouvons expliquer cette différence par la figure ci-dessous. En fait, un seul corpus, Relectures d'articles, possède une variabilité de la probabilité a posteriori de phrases qui est souvent plus haute que celle de documents. Cela nous confirme que ses documents sont bien distingués au niveau des phrases.

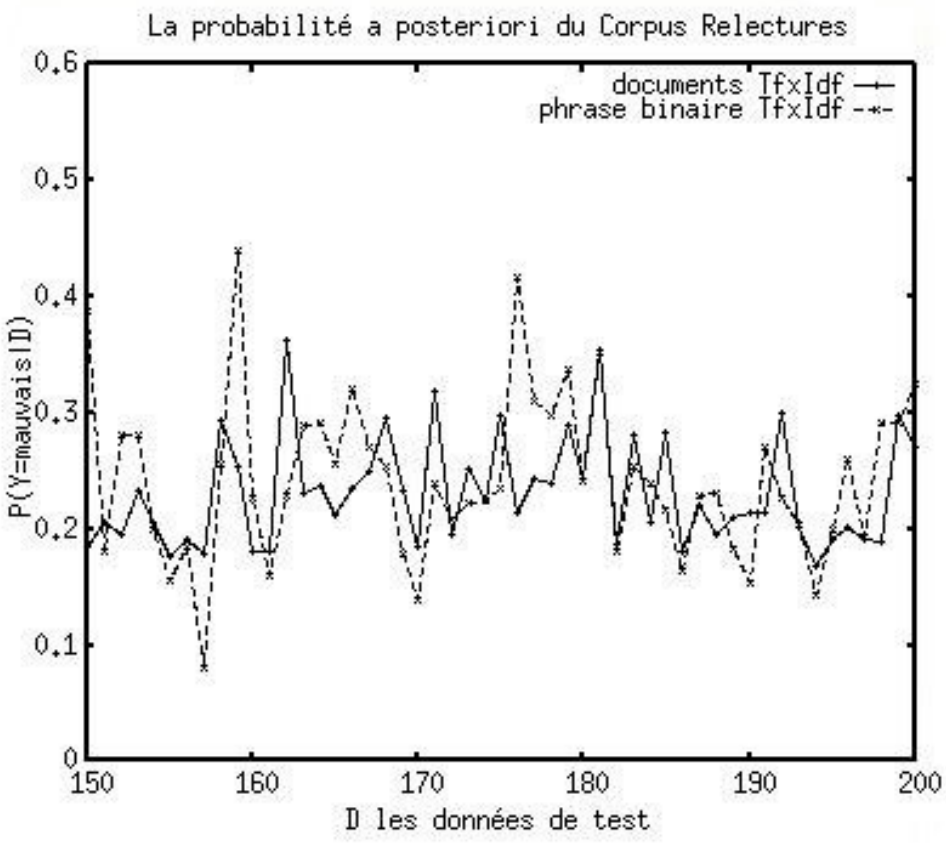

Figure 3 – La probabilité a posteriori du corpus Relectures d'articles

4.2 La réduction de mots dans le sac de mots

Comme nous l'avons précisé dans la section 2.2, nous adoptons pour méthode d'éliminer les mots les plus utilisés de la langue française afin de réduire le nombre de mots à analyser. Il y a 13 mots à éliminer lors des analyses comme suit :

la:75384:15577,*la*:111332:16126,*les*:72674:14753,*un*:36345:12358,

une:34421:11765,*l*:88199:15754,*et*:67158:14702,*Une*:1192:1059,

Un:1144:1021,*Les*:6619:4272,*L*:7312:5122,*La*:9299:5566,*Le*:9534:5851,

Avec mot :nombre_total_de_ce_mot : nombre_de_document_contient_ce_mot dans le corpus débats parlementaires

Ces mots sont éliminés des sacs de mots et naturellement n'apparaissent pas dans les vecteurs de documents ou de phrases. En réalité, l'ensemble des mots les plus utilisés peut très bien augmenter en y ajoutant les prépositions « à » ou « de », pourtant ces prépositions jouent un rôle important dans les expressions telles que « à travers » ou « par rapport à ». Nous constatons le même phénomène dans les cas de « ne .. que » (l'expression de l'unique) ou « ne .. pas » (le négatif), etc. Nous avons choisi d'éliminer les 13 mots ci-dessus pour la raison qu'ils sont plutôt souvent liés au genre du mot (masculin/féminin), donc au mécanisme du vocabulaire, qu'au sens apporté à la phrase. Ainsi l'information de la phrase est conservée le mieux possible. Une autre raison est que, si on élimine un grand nombre de mots, on risque de se retrouver devant une phrase, ou même un document « vide » car aucun mot n'est retenu, comme dans le cas du corpus de relecture d'articles.

Diviser le corpus d'apprentissage 60% et de test 40%	
Corpus d'apprentissage	(A)ccuracy
Critiques de cinéma, spectacles, livres, BD et CD	0,60096
Test de jeux vidéo	0,70767
Relectures d'articles de conférences	0,42372
Débats parlementaires	0,70697

Tableau 8 Les résultats des corpus d'apprentissage avec la réduction de mots

Le nombre de mots ne diminue pas d'une façon brutale quand on compare ce résultat au celui du sac de mots complet (voir le tableau 6). Notre but est de trouver une méthode pour augmenter la valeur d'exactitude (A), mais cette méthode la diminue dans l'ensemble des corpus. C'est pour cette raison que nous ne soumettons pas cette méthode mais la remplaçons par l'analyse des documents en les divisant en phrases comme expliqué ci-dessus.

5 Conclusions

La méthode que nous avons présentée ici repose sur la saisie des informations probabilistes à partir du modèle d'apprentissage discriminant. Nous avons besoin de nous appuyer sur deux principes, le premier est qu'il n'est pas nécessaire de calculer, à chaque itération dans l'algorithme d'apprentissage MVS, une estimation probabiliste. Le deuxième est qu'une étape supplémentaire est réalisée pour rétablir

cette estimation après l'avoir apprise, enfin, nous obtenons les informations désirées.

Corpus de test	(P)récision	(R)appel	(F)score
Critiques de cinéma, spectacles, livres, BD et CD	0,5276 ± 0,0982	0,4829 ± 0,0683	0,5004 ± 0,0668
Test de jeux vidéo	0,6925 ± 0,0996	0,6367 ± 0,0921	0,6604 ± 0,0864
Relectures d'articles de conférences	0,4804 ± 0,0490	0,4614 ± 0,047	0,4706 ± 0,0468
Débats parlementaires	0,6545 ± 0,0564	0,6298 ± 0,0645	0,6416 ± 0,0594

Tableau 9 Les résultats des tests (moyenne, écart-type) de toutes les équipes

En générale, la méthode a donné des résultats supérieurs à la moyenne de l'ensemble des participants de DEFT07 sur les deux corpus de Débats et Critiques. Cependant ce résultat reste inférieur à la moyenne sur les deux autres. La difficulté se situe dans l'augmentation du nombre de phrases lorsque nous divisons chaque document en phrases individuelles en représentant localement le document. Nous avons passé environ deux heures à simplement créer les deux corpus d'apprentissage et de test pour chacun des deux corpus « Test de jeux vidéo » et « Débats parlementaires ». Notre principe de calcul des valeurs est bien confirmé par l'usage, en plus, nous voulons rappeler un inconvénient de cette méthode, il faut résoudre le problème (8) à chaque donnée d'entrée (voir la sélection 3.1.3.b). Le problème (8) semble correspondre à la vraisemblance globale après la discrimination locale qui a été créée par la MVS. Nous désirons appliquer ces résultats aux modèles d'apprentissage probabiliste et espérons que les résultats vont s'améliorer.

Remerciements

Nous remercions sincèrement le comité de DEFT07 pour leur effort d'organisation, de collecte des données ainsi que pour les explications qui nous ont été fournies.

Bibliographie

Salton, G. and McGill, M. J. (1983). Introduction to modern information retrieval. McGraw-Hill, ISBN 0070544840.

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

F. Vallet, P. Réfrégier, J. G. Cailton (1991). Linear discrimination: explicit and iterative solutions. In Pattern recognition and neural networks Vol. 2, Pages: 91 – 114

Kenneth W. Church and William A. Gale (1995). Inverse Document Frequency (IDF): A Measure of Deviation from Poisson. In Proceedings of the Third Workshop on Very Large Corpora,pp 121--130.

Corinna Cortes and V. Vapnik,(1995). Support-Vector Networks. Journal of Machine Learning,vol. 20.

William W. Cohen. (1995) Learning to classify English text with ILP methods. In Luc De Raedt, editor, Advances in ILP. IOS Press.

D. Price, S. Knerr, L. Personnaz, and G. Dreyfus (1995). Pairwise neural network classificateurs with probabilistic outputs. In Neural Information Processing Systems, volume 7, pages 1109-1116. The MIT Press.

T. Hastie and R. Tibshirani, (1996). Classication by pairwise coupling.Technical report, Stanford University and University of Toronto.

N. Friedman and M. Goldszmidt (1996). Building classificateurs using Bayesian networks. In AAAI '96.

Vapnik, V. (1998). Statistical Learning Theory. Wiley-Interscience, New York.

J. Platt (1998). Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, Advances in kernel methods - support vector learning. MIT Press.

J. Platt, (2000).Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in *Advances in Large Margin Classificateurs*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press.

Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schoelkopf, B. (2001). An Introduction to Kernel-Based Learning Algorithms. IEEE transactions on neural networks 12, Nr.2, pp.181-201/ISSN 1045-9227

D. Tax and R. Duin (2002). *Using two-class classificateurs for multi-class classification*. In International Conference on Pattern Recognition, Quebec City, QC, Canada, August.

T. Joachims (2002)**.** Learning to Classify Text using Support Vector Machines, Kluwer Academic Publishers, May 2002,ISBN 0-7923-7679-X.

Sarah Zelikovitz and Haym Hirsh (2002). Integrating Background Knowledge into Nearest-Neighbor Text Classification.*Proceedings of the 6th European Conference on Case Based Reasoning*. Springer Verlag.

H.-T. Lin, C.-J. Lin, and R. C. Weng (2003). A note on Platt's probabilistic outputs for support vector machines. Technical report, Department of Computer Science, National Taiwan University.

T.-F. Wu, C.-J. Lin, and R. C. Weng (2004). Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning 5:975-1005.

Summary

The text classification is simply represented by an estimation of the posterior probability $P(Y/D)$. We shall present a method based on the SVM to realize this task. There are two analyze approaches: the first represent each document as unit, the second tries to divide document into sentences. We have also described a relationship between the local discrimination and the global likelihood.

Keywords: SVM, Probability Estimation, Text Classification, Multi-class classification.