De nouvelles pondérations adaptées à la classification de petits volumes de données textuelles

Flavien Bouillot, Pascal Poncelet, Mathieu Roche

De nouvelles pondérations adaptées à la classification de petits volumes de données textuelles

Flavien Bouillot[*,**], Pascal Poncelet[*], Mathieu Roche[*,***]

[*] LIRMM, Univ. Montpellier 2, CNRS – France
[**]ITESOFT, Aimargues – France
[***]TETIS, Cirad, Irstea, AgroPariTech – France
Prenom.Nom@lirmm.fr

Résumé. Un des défis actuels dans le domaine de la classification supervisée de documents est de pouvoir produire un modèle fiable à partir d'un faible volume de données. Avec un volume conséquent de données, les classifieurs fournissent des résultats satisfaisants mais les performances sont dégradées lorsque celui-ci diminue. Nous proposons, dans cet article, de nouvelles méthodes de pondérations résistant à une diminution du volume de données. Leur efficacité, évaluée en utilisant des algorithmes de classification supervisés existants (Naive Bayes et Class-Feature-Centroid) sur deux corpus différents, est supérieure à celle des autres algorithmes lorsque le nombre de descripteurs diminue. Nous avons étudié en parallèle les paramètres influençant les différentes approches telles que le nombre de classes, de documents ou de descripteurs.

1 Introduction

La classification supervisée de documents vise à déterminer la ou les catégories potentielles d'un document à partir de son contenu (les termes le composant). Dans un cadre supervisé, le processus se décompose généralement en 2 phases :

1. la phase d'apprentissage, qui vise à créer un modèle à partir d'un ensemble d'exemples étiquetés (documents dont la classe est connue),
2. la phase de classification, qui va déterminer la ou les catégories d'un document dont la classe est inconnue par application du modèle.

Bien entendu, la qualité du modèle dépend de la qualité et du nombre d'exemples disponibles. Ainsi plus il y a d'exemples, plus les observations seront fiables et plus le modèle sera précis et efficace.

Il peut cependant s'avérer intéressant de pouvoir élaborer un modèle de classification fiable à partir d'un faible nombre de descripteurs (Forman et Cohen, 2004). Par exemple, le développement des réseaux sociaux, avec un nombre de plus en plus important de messages en temps réel mais d'une taille limitée (comme un tweet limité à 140 caractères), implique la mise à disposition d'outils capables de les classer rapidement avec un volume restreint de données. Dans ce contexte, l'extraction de descripteurs pertinents et discriminants représente un défi

intéressant. De même, il peut arriver que le nombre d'exemples utiles à la création du modèle lors de la phase d'apprentissage soit lui même très limité. De plus, déterminer la classe d'un document est une opération longue qui, généralement, nécessite un expert du domaine. Certaines approches se fondent sur un nombre restreint d'exemples, les approches de classification semi-supervisées qui utilisent les documents non étiquetées pour compléter l'apprentissage supervisé ou encore d'apprentissage actif qui consiste à construire l'ensemble d'apprentissage du modèle de manière itérative, en interaction avec un expert humain. Certaines de ces méthodes appliquées à un faible nombre d'exemples sont présentées dans (Zeng et al., 2003; Lin et Cohen, 2010) mais elles impliquent d'avoir un grand nombre de documents à disposition pour améliorer le modèle ce qui n'est pas toujours possible.

La plupart des algorithmes actuels de classification supervisée de documents nécessitent un nombre suffisant d'exemples pour créer un classifieur performant et les performances décroissent en même temps que le nombre d'exemples diminue. Notre principale contribution est de proposer de nouvelles pondérations de descripteurs textuels pour traiter des jeux d'exemples de petite taille. Ces méthodes de pondérations ont été intégrées dans des approches classiques de classification (Class Feature Centroid et Naive Bayes).

L'article est organisé de la manière suivante : dans la section 2, nous discutons de l'intérêt de nouvelles pondérations. Des mesures permettant d'extraire les descripteurs les plus pertinents d'une classe sont présentées en section 3. L'intégration de ces mesures au sein d'algorithmes de classification en apprentissage supervisé est décrite dans la section 4. Les résultats expérimentaux, comprenant notamment une comparaison avec d'autres types d'approches, sont proposés en section 5. Enfin, la section 6 conclue et présente quelques perspectives.

2 Proposition

La classification supervisée de documents cherche à déterminer la catégorie (ou classe) d'un document à partir de son contenu. Considérons $C = C_1, C_2, ..., C_n$ un ensemble de n classes et $D = d_1, d_2, ..., d_m$ un ensemble de m documents. Chaque document est rattaché à une classe et nous notons $D_{i,j}$ le i^{eme} document de la classe j et $D_j = d_{1,j}, d_{2,j}, d_{m,j}$, l'ensemble des documents de la classe j.
Les documents sont représentés selon le modèle sac de mots (*bag of words*) de Salton (Salton et McGill, 1986) et tous les termes de tous les documents forment un dictionnaire (ensemble des termes apparaissant au moins une fois dans la collection de documents). $L = t_1, t_2, ..., t_{|L|}$ est un dictionnaire qui contient $|L|$ termes où t_i ($1 \leq i \leq |L|$) est un terme unique dans le dictionnaire. Chaque document est représenté par un vecteur pondéré de termes sans indication de position dans le document. $\overrightarrow{D_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ est la représentation vectorielle du document j où w_{ij} est le poids (ex : Fréquence, Booléen,...) du terme t_i pour le document j.
Le poids w_{ij} d'un terme dépend à la fois de son poids intra-classe et de son poids inter-classes. Le poids **intra-classe** permet d'évaluer l'importance du terme au sein de la classe (*Est-ce un terme représentatif de la classe ?*) alors que le poids **inter-classes** mesure pour le terme, le pouvoir discriminant de la classe par rapport aux autres classes (*Est-ce que le terme représente toutes les classes ?*). Des méthodes de pondérations sont présentées dans (Guan et al., 2009) et (Zhang et al., 2012). Ces pondérations, dérivées du $TF\text{-}IDF$, reposent à la fois sur la fréquence d'un terme dans les documents au sein d'une classe mais aussi sur la présence ou

l'absence de ce terme au sein des autres classes. Nous pensons que deux situations posent des problèmes :

- En présence de classes composées d'un petit nombre de documents, l'impact de la fréquence de document est trop importante.
- En présence de classes sémantiquement proches puisqu'un terme est considéré comme représentant une classe même s'il n'apparaît qu'une seule fois dans la classe. Dans ce cas, le nombre d'occurrences du terme au sein des classes n'est pas pris en compte.

De plus, ces pondérations ne sont appliquées que dans une approche de type Class-Feature-Centroid quand nous pensons qu'elles peuvent être pertinentes dans d'autres approches de classification en apprentissage supervisé.

Dans la section suivante, nous proposons de nouvelles pondérations pour répondre à ces problèmes.

3 Nouvelles pondérations

Nous présentons dans cette section de nouvelles méthodes de pondérations intra-classes (Section 3.2) et inter-classes (Section 3.3) dérivées du TF-IDF. Nous commençons par rappeler les principes de base du TF-IDF.

3.1 Les principes du TF-IDF

Le principe du TF-IDF est de donner un poids plus important aux termes les plus spécifiques d'un document (Salton et McGill, 1986). Le TF-IDF est une méthode de pondération éprouvée dont l'efficacité a été démontrée à de nombreuses reprises. Cette mesure repose sur le produit entre la fréquence du terme (TF : *Term-frequency*) et la fréquence inverse du document (IDF : *Inverse Document Frequency*). La fréquence du terme correspond au nombre d'occurrences d'un terme dans un document et représente le poids du terme au sein du document, aussi appelé poids intra-document. Pour un document d_j et un terme t_i, la fréquence du terme est calculée par :

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

où $n_{i,j}$ correspond au nombre de fois où le terme t_i apparait dans le document d_j et où le dénominateur correspond au nombre total de termes du document d_j. La fréquence inverse de document (*IDF*) mesure l'importance du terme dans le corpus. L'objectif est de donner un poids plus important aux termes qui apparaissent dans peu de documents. Il s'agit du poids inter-documents qui est calculé en considérant le logarithme de l'inverse de la fréquence de documents qui contiennent le terme dans le corpus :

$$IDF_i = log_2 \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

avec $|D|$, le nombre de documents du corpus et $|\{d_j : t_i \in d_j\}|$ le nombre de documents qui contiennent le terme t_i.

Le TF-IDF est obtenu en multipliant les poids intra-document et inter-documents du terme t_i pour le document d_j : $TF\text{-}IDF_{i,j}$=$TF_{i,j} \times IDF_i$.

Pour évaluer la représentativité d'un terme dans une classe et non dans un document, nous proposons une mesure adaptée sur les poids intra-classes et inter-classes d'un terme.

3.2 Poids Intra-classe

Dans un premier temps, nous proposons une méthode de pondération fondée sur la fréquence de documents du terme dans la classe comme décrit dans (Guan et al., 2009), appelé $inner\text{-}weight^{Df}$. Nous calculons le poids $inner\text{-}weight_{ij}^{Df}$ du terme $t_i$de la classe j selon la formule (1). Nous considérons que le terme le plus représentatif d'une classe n'est pas nécessairement le terme le plus fréquemment utilisé dans la classe mais celui utilisé dans le plus grand nombre de documents de la classe. De plus, une telle mesure peut se révéler particulièrement pertinente pour le traitement de documents de longueurs déséquilibrées. En effet, dans ce cas, les descripteurs linguistiques présents dans les documents de plus grandes tailles auront un impact similaire aux termes présents dans les plus petits documents.

$$inner\text{-}weight_{ij}^{Df} = \frac{DF_{ti}^{j}}{|d_j|} \tag{1}$$

Avec :
- DF_{ti}^{j} : Nombre de documents contenant le terme t_i dans la classe C_j
- $|d_j|$: Nombre de documents dans C_j

Néanmoins $Inner\text{-}weight^{Df}$ est confronté aux mêmes limites que celles présentées dans la Section 2 avec les classes composées d'un faible nombre de documents, à savoir l'impact de la fréquence de documents dans les classes les moins pourvues sera disproportionné par rapport aux classes les plus fournies. Ainsi, nous proposons une autre méthode de pondération fondée sur le terme plutôt que sur le document. Nous considérons la fréquence du terme plutôt que la fréquence du document. Cette méthode est appelée $inner\text{-}weight^{Tf}$. Nous définissons le poids $inner\text{-}weight_{ij}^{Tf}$ du terme t_i de la classe j selon la formule (2).

$$inner\text{-}weight_{ij}^{Tf} = \frac{TF_{ti}^{j}}{|n_j|} \tag{2}$$

Avec :
- TF_{ti}^{j} : Nombre d'occurrences du terme t_i dans la classe C_j
- $|n_j|$: Nombre de termes total dans la classe C_j

Dans cette section, nous avons redéfini deux pondérations intra-classes appelées $inner\text{-}weight^{Tf}$ et $inner\text{-}weight^{Df}$. Dans la section suivante, nous nous intéressons à la pondération inter-classes.

3.3 Poids Inter-classes

Nous proposons une première méthode que nous nommons $inter\text{-}weight^{class}$ définie par la formule (3). Cette méthode, fondée sur le nombre de classes qui contiennent le terme, a été utilisée dans (Guan et al., 2009).

$$inter\text{-}weight_{ij}^{class} = log_2 \frac{|C|}{C_{ti}} \tag{3}$$

Avec :

- $|C|$: Nombre de classes
- C_{ti} : Nombre de classes contenant le terme t_i

Cependant, nous estimons que cette approche, qui consiste à considérer la présence ou l'absence d'un terme dans une classe sans prendre en compte la fréquence, est trop restrictive comme par exemple dans les cas ci-dessous :

- *Présence de peu de classes* : Plus le nombre de classes diminue, plus l'influence de poids inter-classes sera importante dans le poids final.
- *Présence de classes sémantiquement proches* : Des classes proches sémantiquement vont partager un grand nombre de termes en commun mais à des fréquences potentiellement différentes.
- *Présence d'un grand nombre de termes par classe* (lié à un grand nombre de documents ou à des documents très longs) : Plus le nombre de termes est important, plus la probabilité qu'un terme apparaisse au moins une fois par classe est forte.

Suite à ces observations, nous suggérons de considérer l'absence ou la présence d'un terme dans les documents en lieu et place des classes. Seuls les documents des autres classes doivent être pris en compte. En effet la représentativité d'un terme au sein de la classe étudiée est prise en compte dans la pondération intra-classe et considérer l'ensemble des documents du corpus, consisterait à prendre doublement en compte la classe étudiée. Pour cela nous définissons le poids $inter\text{-}weight^{doc}$ selon la formule (4).

$$inter\text{-}weight_{ij}^{doc} = log_2 \frac{|d \notin C_j| + 1}{|d : t_i \notin C_j| + 1} = log_2 \frac{|d| - |d \in C_j| + 1}{|d : t_i| - |d : t_i \in C_j| + 1} \tag{4}$$

Avec :

- $|d \notin C_j|$: Nombre de documents n'appartenant pas à la classe C_j
- $|d : t_i \notin C_j|$: Nombre de documents n'appartenant pas à la classe C_j qui contient t_i
- $|d|$: Nombre de documents dans l'ensemble des classes
- $|d \in C_j|$: Nombre de documents de la classe C_j
- $|d : t_i|$: Nombre de documents dans l'ensemble des classes contenant le terme t_i
- $|d : t_i \in C_j|$: Nombre de documents de la classe C_j qui contient t_i
- En ajoutant 1, permet de prévenir le cas où t_i est uniquement utilisé dans C_j (quand $|d : t_i \notin C_j| = |d : t_i| - |d : t_i \in C_j| = 0$)

Pour évaluer nos propositions de pondérations, nous les avons intégrées à des méthodes d'apprentissage supervisé détaillées en section 4.

4 Utilisation des mesures dans un contexte d'apprentissage supervisé

SVM (Support Vector Machine) et Naive Bayes sont considérés parmi les algorithmes de classification supervisée les plus performants. Cependant, ils ne sont pas toujours adaptés en présence de faibles volumes de données (Kim et al., 2006). Nos nouvelles méthodes de pondération peuvent être utilisées dans des approches Naives Bayes ou Class-Features-Centroid qui présentent les avantages suivants : (i) leur facilité de mise en œuvre les rendent bien adaptées pour la classification automatique de documents ; (ii) elles sont toutes deux fondées sur des descripteurs pondérés au niveau des classes, ce qui rend nos mesures bien adaptées à ce type d'algorithme ; (iii) enfin les modèles obtenus sont faciles à interpréter et à valider pour un utilisateur final.

4.1 Nouvelles mesures et classification Class-Feature-Centroid

L'approche Class-Feature-Centroid est un modèle récent présenté dans (Guan et al., 2009). Chaque classe est considérée selon le modèle vectoriel de Salton (Salton et McGill, 1986), sur la représentation en *sac de mots*. Chacune des classes est représentée par un vecteur de termes. $\overrightarrow{C_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ est la représentation de la classe j où w_{ij} est le poids du terme t_i pour la classe j. Lors de la phase de classification d'un document non étiqueté d, le document est aussi considéré comme un vecteur de termes ($\overrightarrow{d} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$) et une distance ou une similarité (ex. cosinus) entre le vecteur du document $\overrightarrow{d}$ et chacun des vecteurs de classe $\overrightarrow{C_j}$ est calculée.

Tout d'abord nous calculons une représentation vectorielle de chaque classe, $\overrightarrow{C_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ pour les différentes combinaisons de poids intra-classes et inter-classes définis précédemment :

- $w_{ij}^{Tf-Class}$=$inner\text{-}weight^{Tf} \times inter\text{-}weight^{class}$
- $w_{ij}^{Df-Class}$=$inner\text{-}weight^{Df} \times inter\text{-}weight^{class}$
- w_{ij}^{Tf-Doc}=$inner\text{-}weight^{Tf} \times inter\text{-}weight^{doc}$
- w_{ij}^{Df-Doc}=$inner\text{-}weight^{Df} \times inter\text{-}weight^{doc}$

Ensuite nous appliquons le modèle $Class - Feature - Centroid$ et nous appelons $Cfc^{Tf-Class}$ l'expérimentation visant à utiliser le poids $w_{ij}^{Tf-Class}$ avec une approche Class-Feature-Centroid (resp $Cfc^{Df-Class}$, Cfc^{Tf-Doc} et Cfc^{Df-Doc}). Comme pondération du vecteur du document non étiqueté, nous choisissons une valeur booléenne [1] indiquant l'absence ou la présence du terme dans le document et nous calculons un produit scalaire entre le vecteur du document et les vecteurs de classes.

1. Le booléen est préféré au $TF\text{-}IDF$ car le $TF\text{-}IDF$ est impacté par les classes les plus grandes et les termes des classes majoritaires présents dans un document non étiqueté seraient sous estimés.

4.2 Nouvelles mesures et Naive Bayes

Nous avons aussi intégré nos pondérations dans une approche Naive Bayes. Le classifieur Naive Bayes est un classifieur de type probabiliste défini dans (Lewis, 1998) très utilisé pour la classification de texte car il donne de bons résultats malgré l'hypothèse rarement vérifiée d'indépendance conditionnelle à la classe des descripteurs. La probabilité qu'un document non étiqueté (d) composé de i termes t_i appartienne à la classe C_j est donnée par $P(d \in C_j) = P(C_j) \prod_i P(t_i|C_j)$.

Après avoir calculé $C_j = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ où $w_{i,j}$ est le poids du i^{eme} terme de la classe C_j, nous estimons la probabilité qu'un document non étiqueté d appartienne à la classe C_j : $P(d \in C_j) = P(C_j) \prod_i (w_{i,j} + 1)$. Ajouter 1 permet de prévenir le cas où le terme n'apparait pas dans la classe, quand la probabilité vaut zéro. Les expérimentations sont appelées $Nb^{Tf-Class}$, $Nb^{Df-Class}$, Nb^{Tf-Doc} et Nb^{Df-Doc} dans la suite du document.

4.3 Algorithmes de comparaison

Dans la section suivante (section 5) nous comparons ces huit méthodes de classification supervisée (4 pondérations pour chacun des 2 algorithmes) avec différents algorithmes implémentés dans Weka [2] :

- Deux implémentations de SVM, *SMO*, qui utilise un noyau polynomial (Platt, 1999) et *LibSVM*, qui utilise un noyau linéaire (Chang et Lin, 2011).
- Une implémentation Naïve Bayes, *DMNB* (Su et al., 2008).
- Un arbre de décision, LadTree, (Holmes et al., 2002).

D'autres expérimentations ont été réalisées, toujours avec Weka (NaiveBayes (John et Langley, 1995), NaiveBayes Multinomial (McCallum et al., 1998), LibSVM avec fonction à base radial et LibSVM avec noyau polynomial (Chang et Lin, 2011), J48 et RepTree(Quinlan, 1993)). Les résultats étant moins satisfaisants, ils ne sont pas présentés dans cet article [3].

Chacun des algorithmes a été testé en validation croisée (*3-fold cross validation*) et les résultats indiqués ci-après sont la moyenne de ces 3 itérations. Le nombre de 3 itérations a été retenu afin d'avoir un nombre suffisant de données en apprentissage et test. Les différentes expérimentations fondées sur ces approches sont détaillées dans la section suivante.

5 Expérimentations

5.1 Protocole expérimental

Nous avons évalué nos propositions sur 2 corpus différents :

- Le corpus *Reuters-21578* [4], qui est fréquemment utilisé par la communauté pour évaluer la qualité des modèles, est un ensemble de dépêches écrites en anglais mises à disposition par l'agence Reuters. Les news sont regroupées dans différentes catégories comme par exemple "sucre", "huile" ou "or", etc. Les documents ont été étiquetés manuellement.
- Le corpus *Tweet* est composé de tweets de langue française émis durant les campagnes présidentielle et législative française de 2012. Nous avons collecté les messages au cours

2. http ://www.cs.waikato.ac.nz/ml/weka/
3. ils sont disponibles à l'adresse `www.lirmm.fr/~bouillot/weipond`
4. http ://trec.nist.gov/data/reuters/reuters.html/

TAB. 1 – *Corpus*

Corpus	Classes	Documents	Termes	Termes distincts
Reuters-21578	39	14 701	1 237 264	59 281
Tweet	5	1 186	1 579 374	16 593

du projet Polop [5]. Le corpus est composé de plus de 1 million de tweets provenant de 213 005 utilisateurs. Nous considérons pour ce corpus qu'un parti politique est une classe et qu'un document est l'ensemble des tweets émis par un même utilisateur.

Pour chaque corpus, nous avons supprimé les mots outils (stopwords) et les termes inférieurs à 3 caractères et, pour des raisons de performance et de fiabilité d'interprétation des résultats, nous avons conservé les classes composées à minima de 45 documents (30 documents en apprentissage et 15 en test). Cela représente 39 catégories pour le corpus *Reuters-21578* et 5 principaux partis politiques français (Parti Socialise, UMP, Modem, EELV et Front de Gauche) [6] pour le corpus *Tweet*. Le Tableau 1 présente les caractéristiques des 2 corpus après pré-traitement.

5.2 Résultats

La qualité des modèles est évaluée par la Précision, le Rappel et la F-mesure (moyenne harmonique de la Précision et du Rappel). Nous avons évalué la Micro-Moyenne (calcul global du Rappel et de la Précision sur l'ensemble des classes) et la Macro-Moyenne (calcul du Rappel et de la Précision pour chaque classe, puis calcul de la moyenne des classes (Nakache et Metais, 2005)).

Pour étudier le comportement de nos pondérations au sein des approches Naive Bayes et Class-Feature-Centroid et pour comparer nos résultats avec les autres algorithmes de classification, nous avons réalisé plusieurs expérimentations sur les corpus *"Reuters-21578"* et *"Tweet"*. Nous avons étudié l'impact du *nombre de termes*, du *nombre de documents* et du *nombre de classes* sur les résultats de classification.

5.2.1 Conséquences du nombre de termes sur la classification

Tout d'abord, sur le corpus *Tweet*, nous avons fixé le nombre de classes (5) et de documents (1186) et décidé de supprimer aléatoirement des termes afin de diminuer le **nombre de termes par document**. Nous avons réalisé sept expérimentations résumées dans le Tableau 2.

Nous présentons l'évolution de la F-mesure en fonction des expérimentations dans le Tableau 3. Les Macro-Moyenne et Micro-Moyenne suivant la même tendance, nous ne reproduisons ici que la Micro-Moyenne. Ces expérimentations nous permettent de tirer 2 enseignements : (1) les nouvelles pondérations intégrées aux approches Naive Bayes et Class-Feature-Centroid donnent des résultats meilleurs que les algorithmes SVM et DMNB quand le nombre de termes est faible (expérimentations 6 et 7), (2) elles donnent de meilleurs résultats que les algorithmes LadTree et LibSVM dans tous les cas.

5. `http://www.lirmm.fr/~bouillot/polop`
6. Le Front National, dû au faible nombre d'élus actifs sous Twitter, était sous représenté dans le corpus.

TAB. 2 – *Expérimentation 1 : 7 expérimentations sur le corpus Tweet*

Itération	Termes	Moyenne termes par doc	Termes distincts
1	1 579 374	1332	16 593
2	1 322 148	1115	15 993
3	613 777	518	13 441
4	264 025	223	10 633
5	202 166	175	9 803
6	157 177	133	9 029
7	76 851	66	7 007

TAB. 3 – *Expérimentation 1 : évolution F-mesure (Micro-Moyenne)*

Algo	1	2	3	4	5	6	7
DMNB	**93%**	**92%**	**87%**	**77%**	**78%**	70%	60%
LadTree	72%	72%	67%	56%	56%	53%	49%
LibSVM	67%	22%	30%	51%	50%	51%	51%
SMO	**91%**	**90%**	**82%**	**71%**	70%	61%	51%
$Cfc^{Df-Class}$	79%	79%	71%	57%	57%	57%	53%
Cfc^{Df-Doc}	38%	38%	37%	37%	37%	37%	38%
$Cfc^{Tf-Class}$	**86%**	**86%**	**82%**	**72%**	**72%**	**72%**	**67%**
Cfc^{Tf-Doc}	57%	56%	55%	52%	52%	52%	50%
$Nb^{Df-Class}$	80%	79%	71%	57%	55%	53%	47%
Nb^{Df-Doc}	37%	37%	37%	37%	37%	38%	38%
$Nb^{Tf-Class}$	**88%**	**87%**	**83%**	**74%**	**71%**	**68%**	**59%**
Nb^{Tf-Doc}	57%	56%	55%	52%	51%	50%	46%

5.2.2 Conséquences du nombre de classes sur la classification

Ensuite, sur le corpus *"Reuters-21578"*, nous nous intéressons à l'impact du **nombre de classes**. Nous fixons le nombre de documents par classe (50) et nous supprimons des classes pour passer de 28 à 2 classes. Nous avons réalisé 7 expérimentations, présentées dans le Tableau 4.

Comme nous pouvions le supposer, les algorithmes ont un meilleur comportement en présence d'un nombre limité de classes. Les algorithmes suivent la même tendance et nous décidons de ne pas reproduire ici le détail des résultats [7]. Nous pouvons préciser que LadTree est plus impacté par un grand nombre de classes que les autres et que les nouvelles pondérations intégrées dans des approches Naive Bayes ou Class-Feature-Centroid donnent des résultats lé-

7. Les résultats sont disponibles à l'adresse `www.lirmm.fr/~bouillot/weipond`

TAB. 4 – *Expérimentation 2 : 7 expérimentations sur le corpus Reuters-21578*

Itération	Classes	Documents	Nb moyen de documents par classe	Termes distincts
1	28	1 398	50	164 540
2	25	1 248	50	148 693
3	20	998	50	124 736
4	15	748	50	97 517
5	10	499	50	62 808
6	5	250	50	31 101
7	2	100	50	10 508

TAB. 5 – *Expérimentation 3 : 9 expérimentations sur le corpus Reuters-21578*

Itération	Classes	Documents	Nb moyen de documents par classe	Termes distincts
1	10	500	50	62808
2	10	450	45	57 336
3	10	390	39	47 753
4	10	330	33	42 219
5	10	270	27	33 572
6	10	210	21	26 040
7	10	150	15	17 596
8	10	90	9	9 641
9	10	30	3	3 023

TAB. 6 – *Expérimentation 3 : évolution F-mesure (Micro-Moyenne)*

Algo	1	2	3	4	5	6	7	8	9
DMNB	76%	76%	76%	74%	71%	68%	67%	54%	38%
LadTree	**80%**	**80%**	**81%**	**80%**	**79%**	**76%**	**78%**	51%	16%
LibSVM	69%	71%	66%	59%	54%	47%	45%	30%	21%
SMO	73%	72%	71%	68%	64%	59%	57%	41%	22%
$Cfc^{Df-Class}$	**78%**	**79%**	**76%**	73%	72%	72%	69%	56%	36%
Cfc^{Df-Doc}	75%	75%	73%	71%	70%	72%	67%	55%	36%
$Cfc^{Tf-Class}$	**77%**	**78%**	**78%**	**77%**	**78%**	**75%**	**72%**	**64%**	**45%**
Cfc^{Tf-Doc}	**77%**	**78%**	**77%**	**76%**	**77%**	**75%**	**70%**	**63%**	**45%**
$Nb^{Df-Class}$	**77%**	**77%**	**74%**	**72%**	70%	69%	66%	53%	36%
Nb^{Df-Doc}	73%	72%	71%	69%	67%	68%	65%	51%	36%
$Nb^{Tf-Class}$	**78%**	**78%**	**78%**	**77%**	**78%**	**75%**	**71%**	**65%**	**49%**
Nb^{Tf-Doc}	**77%**	**78%**	**77%**	**76%**	**77%**	**75%**	**71%**	**64%**	**49%**

gèrement meilleurs que ceux des autres approches (des observations similaires sont présentées ci-après). Il est aussi intéressant de noter que les résultats sont similaires sur un corpus de langue française (*Tweet*) et un corpus de langue anglaise (*Reuters-21578*).

5.2.3 Conséquences du nombre de documents sur la classification

Dans la troisième expérimentation, nous nous concentrons sur l'étude de l'impact du **nombre de documents par classes**. Nous fixons le nombre de classes (10) et nous diminuons le nombre de documents par classes de 50 à 3. Neuf expérimentations ont été réalisées et elles sont résumées dans le Tableau 5. Comme les classes sont équilibrées, les Micro-Moyenne et Macro-Moyenne sont similaires et nous décidons de ne présenter que l'évolution de la F-mesure pour la Micro-Moyenne dans le Tableau 6.

A partir de ces expérimentations, nous pouvons conclure que les méthodes de pondérations proposées dans cet article intégrées dans des approches Naive Bayes ou Class-Feature-Centroid (1) sont légèrement meilleures que les autres algorithmes (seulement devancé par LadTree) , (2) sont plus résistantes que la plupart des algorithmes quand le nombre de documents disponibles en apprentissage diminue fortement.

6 Conclusions et perspectives

La classification d'un faible volume de documents textuels reste une problématique d'actualité. En effet, même si le nombre de documents ne cesse de croître, il existe de plus en plus de domaines d'applications où nous devons rapidement et à partir d'un faible volume être capable de classer. Par exemple les tweets nécessitent, pour être suivis, de pouvoir classer en temps réel l'information disponible. Attendre d'avoir un nombre suffisant d'éléments n'est pas toujours possible et le décideur souhaite avoir rapidement les documents classés.
Dans cet article, nous avons proposé de nouvelles mesures particulièrement adaptées aux faibles volumes de données (dû à un faible nombre de documents ou à un faible nombre de descripteurs). Nous avons également montré comment ces mesures pouvaient être prises en compte dans un cadre supervisé notamment via Naive Bayes et une approche basée sur les centroides. Les expérimentations menées sur un corpus de tweets et un benchmark plus traditionnel ont permis de montrer que nos nouvelles mesures ont un meilleur comportement que les autres approches lors de la manipulation d'un faible volume de documents. Simples à mettre en place, elles sont tout à fait adaptées à la manipulation de données évoluant rapidement comme les tweets. Notre proposition est fondée sur une extension de la mesure de $TF\text{-}IDF$. Nos travaux actuels évaluent la possibilité de prendre en compte d'autres mesures comme par exemple $OKAPI_{BM25}$(Robertson et al., 1999) afin de mieux appréhender l'impact sur non seulement de petits volumes mais également des documents courts. Dans un contexte de classification, la proximité sémantique des classes est difficile à prendre en compte. Via nos propositions et notamment les mesures inter et intra classes, nous avons montré qu'elles sont adaptées aux faibles volumes. Nous souhaitons à présent proposer à l'utilisateur de pouvoir mieux pondérer ces mesures en fonction de la distribution et du volume de termes.

Références

Chang, C.-C. et C.-J. Lin (2011). Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*(3), 27 :1–27 :27.

Forman, G. et I. Cohen (2004). Learning from little : comparison of classifiers given little training. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, New York, NY, USA, pp. 161–172. Springer-Verlag New York, Inc.

Guan, H., J. Zhou, et M. Guo (2009). A class-feature-centroid classifier for text categorization. In *Proceedings of the 18th international conference on World wide web*, WWW '09, New York, NY, USA, pp. 201–210. ACM.

Holmes, G., B. Pfahringer, R. Kirkby, E. Frank, et M. Hall (2002). Multiclass alternating decision trees. In *Proceedings of the 13th European Conference on Machine Learning*, ECML '02, London, UK, UK, pp. 161–172. Springer-Verlag.

John, G. H. et P. Langley (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 338–345. Morgan Kaufmann Publishers Inc.

Kim, S.-B., K.-S. Han, H.-C. Rim, et S.-H. Myaeng (2006). Some effective techniques for naive bayes text classification. *Knowledge and Data Engineering, IEEE Transactions*

on 18(11), 1457–1466.

Lewis, D. D. (1998). Naive (bayes) at forty : The independence assumption in information retrieval. pp. 4–15. Springer Verlag.

Lin, F. et W. W. Cohen (2010). Semi-supervised classification of network data using very few labels. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '10, Washington, DC, USA, pp. 192–199. IEEE Computer Society.

McCallum, A., K. Nigam, et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Volume 752, pp. 41–48. Citeseer.

Nakache, D. et E. Metais (2005). Evaluation : nouvelle approche avec juges. In *INFORSID'05 XXIII e congrès, Grenoble*, pp. 555–570.

Platt, J. C. (1999). Advances in kernel methods. Chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. Cambridge, MA, USA : MIT Press.

Quinlan, J. R. (1993). *C4.5 : programs for machine learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

Robertson, S., S. Walker, M. Beaulieu, et P. Willett (1999). Okapi at trec-7 : Automatic ad hoc, filtering, vlc and interactive track. pp. 199–210.

Salton, G. et M. J. McGill (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA : McGraw-Hill, Inc.

Su, J., H. Zhang, C. X. Ling, et S. Matwin (2008). Discriminative parameter learning for bayesian networks. In *Proceedings of the 25th international conference on Machine learning*, pp. 1016–1023. ACM.

Zeng, H.-J., X.-H. Wang, Z. Chen, H. Lu, et W.-Y. Ma (2003). Cbc : Clustering based text classification requiring minimal labeled data. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, Washington, DC, USA, pp. 443–450. IEEE Computer Society.

Zhang, X., T. Wang, X. Liang, F. Ao, et Y. Li (2012). A class-based feature weighting method for text classification. *Journal of Computational Information Systems 8*(3), 965–972.

Summary

More and more, in text classification tasks, we need to provide a classifier even if the number of documents for the learning step is quite small. In this paper we evaluate the performance of traditional classification methods to better evaluate their limitation when dealing with small amount of documents during the learning phase and we extend the way of weighting features for taking into account the specificities of the data. New weighting methods are evaluated in Class-Feature-Centroid and Naive Bayes approaches on two different datasets. They demonstrate the efficiency of our approach relative to many other supervised learning algorithms to deal with few data or poor content when the number of features is low. We investigate also on parameters influencing algorithms like the numbers of classes, documents or words.