Classification de textes : de nouvelles pondérations adaptées aux petits volumes

Flavien Bouillot

Collège Doctoral du Languedoc-Roussillon

THÈSE

Pour obtenir le grade de

Docteur

Délivré par l'**Université de Montpellier**

Préparée au sein de l'école doctorale **I2S***
Et de l'unité de recherche **UMR 5506**

Spécialité : **Informatique**

Présentée par **Flavien Bouillot**
bouillot@lirmm.fr

Classification de textes : de nouvelles pondérations adaptées aux petits volumes

Doit être soutenue le 16/04/2015 devant le jury composé de :

Eric Gaussier, Professeur, Université Joseph Fourier de Grenoble	Rapporteur
Mohand Boughanem, Professeur, Université Paul Sabatier de Toulouse	Rapporteur
Bruno Cremilleux, Professeur, Université de Caen Basse-Normandie	Examinateur
Vincent Poulain d'Andecy, Directeur de Projets, Société ITESOFT	Invité
Pascal Poncelet, Professeur, Université de Montpellier	Directeur
Mathieu Roche, Chercheur HDR, Cirad & Université de Montpellier	Directeur

* **I2S :** École doctorale Information Structures Systèmes

Résumé

Au quotidien, le réflexe de classifier est omniprésent et inconscient. Par exemple, dans le processus de prise de décision où face à un élément (un objet, un événement, une personne) nous allons instinctivement chercher à rapprocher cet élément d'autres similaires afin d'adapter nos choix et nos comportements. Cette association à telle ou telle catégorie repose sur les expériences passées et les caractéristiques de l'élément. Plus les expériences seront nombreuses et les caractéristiques détaillées, plus fine et pertinente sera la décision. Il en est de même lorsqu'il nous faut catégoriser un document en fonction de son contenu. Par exemple détecter s'il s'agit d'un conte pour enfants ou d'un traité de philosophie. Ce traitement est bien sûr d'autant plus efficace si nous possédons un grand nombre d'ouvrages de ces deux catégories et que l'ouvrage à classifier possède un nombre important de mots.

Dans cette thèse, nous nous intéressons à la problématique de la prise de décision lorsque nous disposons de peu de documents d'apprentissage et que le document possède un nombre de mots limité. Nous proposons pour cela une nouvelle approche qui repose sur de nouvelles pondérations. Elle nous permet de déterminer avec précision l'importance à accorder aux mots composant le document.

Afin d'optimiser les traitements, nous proposons une approche paramétrable. Cinq paramètres rendent notre système adaptable, quel que soit le problème de classification donné. De très nombreuses expérimentations ont été menées sur différents types de documents qui sont de langues variées et en appliquant différentes configurations. Selon les corpus, elles mettent en évidence que notre proposition permet d'obtenir des résultats supérieurs en comparaison avec les meilleures approches de la littérature pour traiter de petits volumes de données.

L'utilisation de paramètres introduit une complexité supplémentaire puisqu'il faut alors déterminer les valeurs optimales. Détecter les meilleurs paramètres et algorithmes est une tâche compliquée dont la difficulté est théorisée à travers le théorème du No-Free-Lunch. Nous traitons cette seconde problématique en proposant une nouvelle approche de méta-classification reposant sur les notions de distances et de similarités. Plus précisément, nous proposons de nouveaux méta-descripteurs adaptés dans un contexte de classification de documents. Cette

approche originale nous permet d'obtenir des résultats similaires aux meilleures approches de la littérature tout en offrant des qualités supplémentaires.

Abstract

Every day, classification is omnipresent and unconscious. For example in the process of decision when faced with something (an object, an event, a person), we will instinctively think of similar elements in order to adapt our choices and behaviors. This storage in a particular category is based on past experiences and characteristics of the element. The largest and the most accurate will be experiments, the most relevant will be the decision. It is the same when we need to categorize a document based on its content. For example detect if there is a children's story or a philosophical treatise. This treatment is of course more effective if we have a large number of works of these two categories and if books had a large number of words.

In this thesis we address the problem of decision making precisely when we have few learning documents and when the documents had a limited number of words. For this we propose a new approach based on new weights. It enables us to accurately determine the weight to be given to the words which compose the document. To optimize treatment, we propose a configurable approach. Five parameters make our adaptable approach, regardless of the classification given problem. Numerous experiments have been conducted on various types of documents in different languages and in different configurations. According to the corpus, they highlight that our proposal allows us to achieve superior results in comparison with the best approaches in the literature to address the problems of small dataset.

The use of parameters adds complexity since it is then necessary to determine optimitales values. Detect the best settings and best algorithms is a complicated task whose difficulty is theorized through the theorem of *No-Free-Lunch*. We treat this second problem by proposing a new meta-classification approach based on the concepts of distance and semantic similarities. Specifically we propose new meta-features to deal in the context of classification of documents. This original approach allows us to achieve similar results with the best approaches to literature while providing additional features.

In conclusion, the work presented in this manuscript has been integrated into various technical implementations, one in the *Weka* software, one in a industrial prototype and a third in the product of the company that funded this work.

Équipe de Recherche

Équipe Advanse, LIRMM

Laboratoire

LIRMM - Laboratoire d'Informatique, Robotique et Micro-Électronique de Montpellier

Adresse

Université Montpellier 2
UMR 5506 - LIRMM
CC477
161 rue Ada
34095 Montpellier Cedex 5 - France

Partenaire

ITESOFT

Société

ITESOFT Groupe

Adresse

Parc Andron
Le Séquoia
30470 Aimargues - France

Table des matières

1 Introduction **1**
1.1 Contexte des travaux 4
1.2 Une double contribution 5
1.3 Organisation de la thèse 5

I Vers une approche de classification supervisée pour petits volumes de données **7**

2 Introduction **9**

3 État de l'art sur les méthodes de pondérations **13**
3.1 La représentation vectorielle 13
3.2 Les méthodes de pondérations non supervisées 16
3.3 Les méthodes de pondérations supervisées 18
3.3.1 Les méthodes binaires 18
3.3.2 Les méthodes non binaires 20
3.4 Discussion 21

4 Vers l'intégration de nouvelles pondérations **25**
4.1 Proposition de nouvelles pondérations 26
4.2 La représentation intra-classe 27
4.2.1 $Intra\text{-}classe^{DF}$ 27
4.2.2 $Intra\text{-}classe^{TF}$ 29
4.3 La représentation inter-classes 30
4.3.1 $Inter\text{-}classes^{ICF}$ 31
4.3.2 $Inter\text{-}classes^{IDF}$ 33
4.3.3 $Inter\text{-}classes^{ITF}$ 34
4.4 Vers une pondération globale 36
4.4.1 Les cinq composants d'une pondération globale 36
4.4.2 La normalisation 39
4.4.3 Mesure globale 42
4.4.4 Mesure globale paramétrique 43
4.5 Bilan et discussions 45

5 Expérimentations de la mesure w_{ij}^{glob} 47
5.1 Corpus Itesoft 47
5.2 Intégration des mesures dans un contexte d'apprentissage supervisé 49
5.2.1 Méthodes basées sur les centroïdes 49
5.2.2 Classifieurs bayésiens 51
5.3 Protocole expérimental 52
5.3.1 Algorithmes de comparaison 52
5.3.2 Critères d'évaluation 53
5.3.3 Paramètres testés 55
5.4 Résultats 55
5.5 Corpus expérimentaux supplémentaires 60
5.5.1 Conséquences du nombre de classes sur la classification . . 64
5.5.2 Conséquences du nombre de documents sur la classification 68
5.5.3 Conséquences du nombre de descripteurs sur la classification 70
5.5.4 Conséquences du déséquilibre entre classes sur la classification 72

6 Bilan et discussions 75

II De nouveaux méta-descripteurs pour représenter un corpus 79

7 Introduction 81

8 État de l'art de la méta-classification 85
8.1 Les algorithmes 86
8.2 Les performances 87
8.3 Les méta-classifieurs 88
8.4 Les problèmes 89
8.5 Les méta-descripteurs 90

9 De nouveaux méta-descripteurs 95
9.1 La similarité comme méta-descripteurs 95
9.2 Le choix de la mesure de similarité 97
9.3 Mesurer les similarités inter-classes et intra-classes 99
9.3.1 Similarités inter-classes 100
9.3.2 Similarités intra-classes 102
9.4 D'un nombre variable de similarités à un nombre fini de méta-descripteurs 106
9.4.1 Application aux similarités inter-classes 108
9.4.2 Application aux similarités intra-classes 109
9.5 Discussion 114

10 Expérimentations avec nos nouveaux méta-descripteurs 117
10.1 Protocole expérimental 118
10.1.1 Méta-descripteurs issus de la littérature 121
10.1.2 Les nouveaux méta-descripteurs 122
10.2 Résultats 123

11 Discussions et conclusions 131

12 Conclusion générale 133
12.1 Synthèse des contributions 133
12.2 Prototypes 134
12.3 Perspectives de recherche 137

Références 143

Chapitre

1

Introduction

Dans la vie de tous les jours, le processus de classification est inconscient et omniprésent.
Inconscient notamment dans un processus de prise de décision [1] où face à un élément (un objet, un événement, une personne...) nous allons instinctivement chercher à rapprocher cet élément à d'autres similaires afin d'adapter nos choix et nos comportements. Par exemple, face à une boisson fumante sortant du micro-onde, notre premier réflexe sera d'attendre que la boisson refroidisse plutôt que de la boire immédiatement. Pour autant sans savoir si cette boisson à cet instant précis serait brulante ou pas, nous rapprochons instinctivement cette situation des expériences passées (vécues ou connues) pour savoir si nous pouvons immédiatement boire le breuvage ou si il est préférable de patienter un petit peu et ce indépendamment que la boisson soit du café, du thé, de la soupe, du lait...

Ce rangement dans une catégorie ("boire tout de suite") ou une autre ("patienter un peu") repose in-fine sur deux éléments :

- Les expériences connues (nous nous sommes déjà brulés avec une boisson sortant du micro-onde ou que nous savons que cela est déjà arrivé à d'autres) ;
- Les caractéristiques de l'élément (ici une boisson fumante sortant d'un micro-onde).

Plus les expériences seront nombreuses et plus les caractéristiques seront détaillées, plus il sera facile de prendre la bonne décision. Si nous chauffons tous les jours 20 centilitres de la même boisson dans la même tasse à la même puis-

1. processus cognitif complexe visant à la sélection d'un type d'action parmi différentes alternatives.

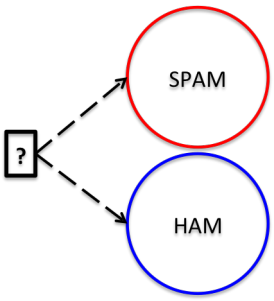


Figure 1.1

sance pendant la même durée dans le même micro-onde, nous sommes capables de dire si notre boisson sera consommable immédiatement ou si il est préférable de patienter. En l'absence de cette connaissance, le risque de prendre la mauvaise décision est plus important. Un enfant découvrant le micro-onde aura tendance à se bruler plus fréquemment qu'un adulte familier de cette expérience.

Parmi les problèmes de classification existants, la classification supervisée de texte est la tâche consistant à attribuer automatiquement une ou plusieurs catégories (ou classes) prédéfinies à un document à partir de son contenu [136, 178]. L'exemple le plus connu est la détection de spams dans les e-mails. Les boites mails à partir de l'objet, du contenu, des adresses vont définir si le message doit être conservé ou mis à l'écart. Par exemple, les e-mails contenant les mots "Viagra" ou "Money" seront généralement orientés vers la catégorie spam.

Nous illustrons cet exemple dans la figure 1.1. Les 2 classes, *SPAM* et *HAM* sont symbolisées par deux cercles de couleur rouge et bleu et le document (l'e-mail) dont nous cherchons à détecter la pertinence est représenté par un rectangle noir.

Pour détecter la classe, nous allons nous baser sur la connaissance apportée par les documents déjà classés (un ensemble d'e-mails dont nous savons qu'il s'agit de SPAM et un ensemble d'e-mails dont nous savons qu'il s'agit de HAM) pour construire un modèle qui permettra de caractériser l'appartenance à une classe. Dans la figure 1.2, ces documents sont symbolisés par un rectangle de couleur.

Ensuite en appliquant ce modèle, nous allons rechercher quelle est la classe d'appartenance la plus probable pour affecter le document (c.f. figure 1.3).

La classification de texte peut en fait être formalisée comme suit :

1. Soit un corpus C composé de i classes (cercle de couleur) où $c_i \in C$ est la $i^{ème}$ classe.
2. Soit D_C un ensemble de documents étiquetés, un document est dit étiqueté si et seulement si une classe $c \in C$ lui est attribuée (rectangles de couleur). D_C^i représente l'ensemble des documents de la classe c_i (rectangle d'une

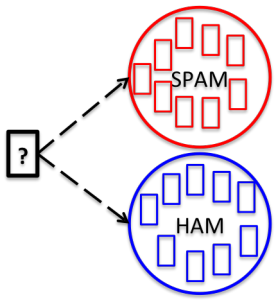


Figure 1.2

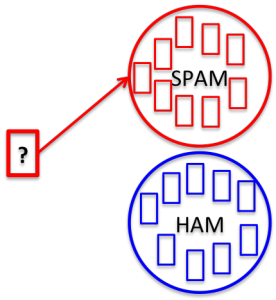


Figure 1.3

même couleur).

3. Soit D_U un ensemble de documents non étiquetés (rectangle noir).
4. La classification de document consiste à attribuer à tous les documents $d \in D_U$ une ou plusieurs classes $c \in C$ en se basant sur l'ensemble des documents étiquetés D_C.

Le processus de classification se décompose généralement en 2 phases [62] :

1. la phase d'apprentissage, qui vise à créer un modèle à partir des documents étiquetés D_C ;
2. la phase de classification, qui détermine pour un document $d \in D_U$, la ou les catégories $c \in C$ les plus probables par application du modèle.

Au cours des 20 dernières années, de nombreuses approches de classification de documents ont été proposées dans la littérature comme les arbres de décisions, les classifieurs bayésiens, les réseaux de neurones, les classifieurs linéaires ou encore les méthodes des centroïdes et des plus proches voisins. Ces différentes approches reposent sur des fondements mathématiques différents et toutes ont prouvé leur efficacité pour traiter un problème donné.

Or devant la grande diversité d'approches ou de paramètres pouvant être considérés comme des candidats crédibles pour répondre à un problème de classification donné, une des grandes difficultés du domaine est de prédire quels seront les

meilleurs parmi les différentes possibilités. Le théorème du *No Free Lunch* [183] stipule que si un classifieur A superforme un classifieur B pour un ensemble de cas donnés, alors il existe autant de cas où B superforme A. La détection de la meilleure approche et/ou des meilleurs paramètres pour un problème de classification donné est une tâche complexe et coûteuse.

1.1 Contexte des travaux

Si les modèles existants offrent généralement de bonnes performances lorsque le nombre documents étiquetés ou le nombre de descripteurs est important, ils sont en revanche reconnus comme étant peu performants lorsque ceux-ci diminuent [74]. Or pouvoir classer à partir d'un faible nombre de documents ou de descripteurs peut aussi être un élément déterminant. 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 est difficile. De même, pouvoir détecter le plus rapidement possible les meilleures approches ou meilleurs paramètres sont des éléments déterminants dans un contexte industriel.

Cette thèse a été réalisée dans le cadre d'un partenariat industriel avec la société ITESOFT. La société ITESOFT, spécialisée dans l'édition et l'intégration de logiciels de traitement automatique des documents papiers et numériques, est confrontée à ce type de problématique.
Pour ses clients, l'objectif est de pouvoir traiter automatiquement le courrier entrant en le redirigeant vers les services appropriés. Par exemple, pouvoir déterminer automatiquement si une lettre reçue concerne le service des "réclamations" ou le service des "souscriptions d'options" [2].

Le nombre de documents labélisés mis à disposition est faible car le processus est coûteux et le nombre de catégories est lui aussi généralement limité (moins de 10 catégories). De plus les domaines d'application potentiels sont vastes (banques, assurances, industries...) et une des difficultés consiste à pouvoir traiter différents corpus très hétérogènes aux caractéristiques différentes. Enfin, une utilisation industrielle (et commerciale) impose des contraintes concernant le temps de traitement du processus dans la mesure où celui se doit de rester le plus rapide possible.
Pour traiter ces problématiques, nous proposons une double contribution que nous présentons dans la section suivante.

2. Par souci de confidentialité les services, les informations et le nom des classes ont été changés.

1.2 Une double contribution

Dans un premier temps, nous nous intéressons à la problématique de classification de documents. La classification de documents repose sur différents éléments, comme les classifieurs, le choix des descripteurs ou encore la pondération des descripteurs. Notre première contribution concerne la pondération des descripteurs adaptée pour la classification de petits volumes. La pondération des descripteurs est une étape cruciale dans le processus de classification. De la qualité de celle-ci dépend la qualité des modèles de classifications. Nous proposons dans ce manuscrit une pondération qui se base sur les différents éléments d'un corpus (i.e. les classes, les documents et les descripteurs) là où les approches de la littérature privilégient toujours l'un au détriment des deux autres. Nous avons défini une pondération paramétrable pour lui permettre de s'adapter au plus près des différents types de corpus, mais en conservant toujours cette volonté de limiter le nombre de documents étiquetés nécessaire.

L'introduction de paramètres nous a permis de développer notre seconde contribution. En s'appuyant sur les conclusions du *No Free Lunch* [183], la détection des paramètres permettant d'obtenir les meilleures performances est une tâche complexe et couteuse en temps. Une recherche pas-à-pas des meilleurs paramètres n'est pas acceptable dans un contexte industriel où le temps de traitement est tout aussi important pour les clients que le nombre de documents étiquetés nécessaire. Une solution pour y parvenir est la méta-classification qui consiste à déterminer les meilleurs paramètres pour un problème donné à partir des expériences réalisées par le passé. Comme la classification, la méta-classification repose sur plusieurs éléments comme les méta-classifieurs ou le choix de méta-descripteurs. Notre seconde contribution concerne une approche originale pour définir de nouveaux méta-descripteurs. Nous étudions la composition des corpus pour définir son empreinte afin de pouvoir rechercher les meilleurs paramètres à partir d'expériences réalisées dans le passé sur des corpus possédant une empreinte similaire.

1.3 Organisation de la thèse

Ce manuscrit est organisé comme suit :

- Dans la première partie de ce manuscrit, nous traitons de la problématique de la pondération des descripteurs (contribution n°1). Dans le Chapitre 3 nous abordons les problématiques liées à la notion de petit volume de document et nous effectuons une revue des solutions proposées dans la littérature. Dans le Chapitre 4, nous présentons une nouvelle pondération de descripteurs adaptée à cette problématique. Nous expérimentons notre approche dans le Chapitre 5 et discutons de cette première contribution dans le Chapitre 6.

- Dans la deuxième partie de ce manuscrit, nous traitons de la définition de nouveaux méta-descripteurs (contribution n°2). Tout d'abord, nous discutons dans le Chapitre 8 de méta-classification et discutons des solutions proposées dans la littérature. Dans le Chapitre 9, nous présentons une approche originale pour décrire un corpus et nous discutons des avantages de notre proposition par rapport aux approches de la littérature. Nous testons l'efficacité de notre approche dans le Chapitre 10 et nous concluons cette seconde partie en discutant de notre proposition dans le Chapitre 11.
- Enfin, nous concluons ce manuscrit en effectuant une synthèse de nos contributions et leurs intégrations dans différents prototypes, et en proposant des extensions à ce travail à court et à moyen terme dans le Chapitre 12.

Première partie

Vers une approche de classification supervisée pour petits volumes de données

Chapitre

2

Introduction

En classification supervisée de document, la qualité du modèle dépend de la qualité et du nombre d'exemples disponibles [39]. 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. Ainsi, plus il y a d'exemples, plus les observations seront fiables et plus le modèle sera précis et efficace.

Comme nous l'avons vu en introduction, il peut cependant s'avérer intéressant de pouvoir élaborer un modèle de classification fiable à partir d'un faible nombre de descripteurs.

Avec un nombre restreint de documents pour créer le modèle, les contraintes sont doubles :

1. Moins il y a de descripteurs dans le corpus en apprentissage, plus il est difficile de définir un poids représentatif de la réalité.
2. Et parallèlement, moins il y a de descripteurs dans le corpus en apprentissage, moins il y a de descripteurs pour prendre la décision et donc plus la précision du poids est importante. Plus précisément, quand il y a de nombreux descripteurs candidats, les approximations vont avoir tendance à se compenser et l'impact de l'imprécision sera limité, alors qu'en présence de peu de descripteurs, ces approximations ne pourront se compenser.

Les approches traditionnelles ne sont pas toujours adaptées en présence de faibles volumes de données [74]. La solution la plus couramment utilisée est la prise en

compte d'information additionnelle récupérée à partir de ressources extérieures (un corpus thématiquement similaire [188] ou plus généralement de sources extérieures comme Wikipédia ou Wordnet [28, 116, 55]). En effet si ajouter de l'information supplémentaire n'est pas pertinent lorsque le nombre de documents disponibles en apprentissage est significatif [8], la probabilité que les descripteurs du document à classer soient absents de l'apprentissage est importante lorsqu'il y a peu d'exemples et l'utilisation de ressources extérieures a été utilisée avec succès dans un grand nombre de travaux [55]. Dans [52], les auteurs utilisent des exemples étiquetés d'autres classes (*inter-class transfer* ou *learning to learn*). Dans [185] les auteurs montrent qu'utiliser deux corpus de langues différentes (chinois et anglais) en complément peuvent améliorer les résultats lorsque le nombre d'exemples positifs est faible.

Dans tous les cas, la généralisation à partir d'un petit nombre d'exemples implique l'introduction *Hypotheses Space Bias* [9] lié au choix et à la provenance des informations additionnelles prises en compte. Par exemple, même si traditionnellement Wordnet [101] est la ressource extérieure la plus utilisée [8, 28, 116, 55], certains auteurs ont remis en cause sa pertinence [107, 150, 174]. De plus il n'est pas toujours possible d'utiliser des ressources extérieures selon les landes utilisées et les domaines de spécialité à prendre en compte.

D'autres approches ont été proposées comme les approches de classification semi-supervisées qui utilisent les documents non étiquetés pour compléter l'apprentissage supervisé [26, 191] ou encore l'apprentissage actif qui consiste à construire l'ensemble d'apprentissage du modèle de manière itérative, en interaction avec un expert humain [30, 145]. Certaines de ces méthodes appliquées à un faible nombre d'exemples sont présentées dans [189, 92] mais elles impliquent d'avoir un grand nombre de documents à disposition pour améliorer le modèle ce qui n'est pas toujours possible.

Dans la suite de ce chapitre, nous considérons que nous nous situons dans le cas d'un faible volume et qui peut par exemple correspondre à l'un de ces scénarios :

1. Dans une approche de classification semi-supervisée ou d'apprentissage actif, lors de la première étape qui consiste à créer un modèle à partir d'un nombre restreint d'exemples.
2. Lors d'un apprentissage en temps réels, où au commencement il n'y a qu'un petit nombre d'exemples disponibles.
3. Dans le cas des sujets émergents où il existe peu de données à disposition pour créer le modèle.

Nous discutons dans le Chapitre 3 des problématiques liées à la notion de petit volume de documents et nous effectuons une revue des solutions proposées dans la littérature. Dans le Chapitre 4, nous présentons une nouvelle pondération de

descripteurs adaptée à cette problématique et nous discutons des avantages de notre proposition par rapport aux approches de la littérature. Nous présentons les différentes expérimentations menées et les discutons dans le Chapitre 5. Le Chapitre 6 conclue cette première partie en rappelant les principaux résultats obtenus.

Chapitre

3 État de l'art sur les méthodes de pondérations

Quelle que soit la méthode de classification retenue, la première opération consiste à représenter les documents de façon à ce qu'ils puissent être traités automatiquement par les classifieurs. La plupart des approches utilisent pour cela la représentation vectorielle des documents [103, 180, 142]. Cette représentation est utilisée dans de nombreux autres domaines connexes de l'apprentissage automatique, par exemple la fouille de texte, la recherche d'information ou le traitement automatique des langues.

Après avoir rappelé la définition de la représentation vectorielle (Section 3.1), nous effectuons une revue des méthodes de pondération utilisées dans la littérature : non supervisées (Section 3.2) puis supervisées (Section 3.3). Nous concluons ce chapitre par une discussion (Section 3.4) concernant les limites des propositions actuelles.

3.1 La représentation vectorielle

La représentation vectorielle ou modèle vectoriel (VSM pour *Vector Space Model*) [135] a été initialement développée pour le système SMART [134]. Le principe consiste à représenter chaque document de la collection comme un point de l'espace, i.e. un vecteur de coordonnées dans l'espace vectoriel [171]. Les coordonnées correspondent en fait aux descripteurs composant le document. Dans la figure 3.1, trois documents sont symbolisés en rouge dans un espace à 3 dimensions (chaque dimension correspondant à un descripteur).
Ainsi, deux points proches dans l'espace vectoriel sont considérés comme séman-

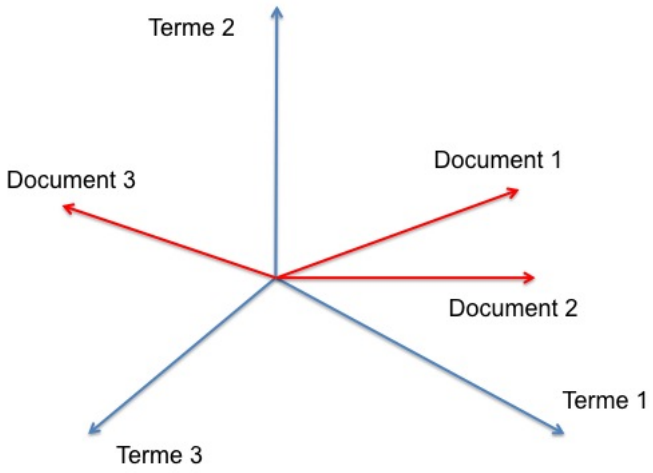


Figure 3.1: Modèle Vectoriel

tiquement similaires alors que deux points distants seront considérés comme sémantiquement différents.

Le modèle vectoriel présente de nombreuses propriétés intéressantes. Outre le fait que la connaissance est extraite automatiquement du corpus et ne nécessite pas de ressources extérieures (ontologie, ressource lexicale), elle permet de prendre en compte les cinq hypothèses suivantes [171] :

- *Statistical semantics hypothesis* : si deux documents ont une représentation vectorielle similaire, ils ont un sens similaire [42].
- *Bag of words hypothesis* : la fréquence d'un descripteur dans un document est un élément important pour mesurer la similarité entre deux documents [135].
- *Distributional hypothesis* : les descripteurs qui apparaissent dans un contexte similaire ont un sens similaire [38, 34].
- *Extended Distributional hypothesis* : des ensembles de descripteurs qui apparaissent fréquemment avec les mêmes descripteurs ont un sens similaire [91].
- *Latent relation hypothesis* : deux descripteurs qui apparaissent dans des groupes de descripteurs similaires ont les mêmes relations sémantiques avec les groupes de descripteurs [170].

Actuellement cette représentation est utilisée par la plupart des moteurs de recherche (la requête utilisateur est projetée dans l'espace [95]), les systèmes de recommandation et de filtrage collaboratif [94] ou encore les algorithmes mesurant les relations sémantiques [169, 109].

Il est courant de représenter un ensemble de vecteurs sous la forme d'une matrice où les lignes représentent les différents vecteurs et les colonnes les différentes coordonnées. Ainsi traditionnellement, pour analyser les similarités, deux types de matrices sont utilisés : Descripteurs - Documents ou Descripteurs - Classes (c.f. figures 3.2 et 3.3).

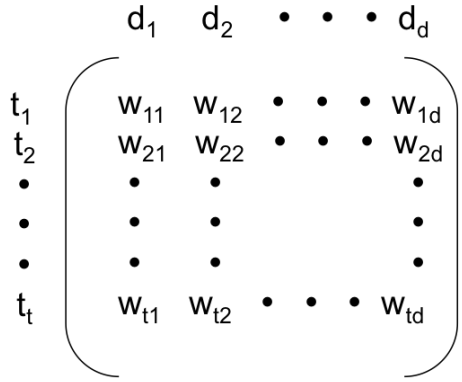


Figure 3.2: Matrice Descripteurs-Documents

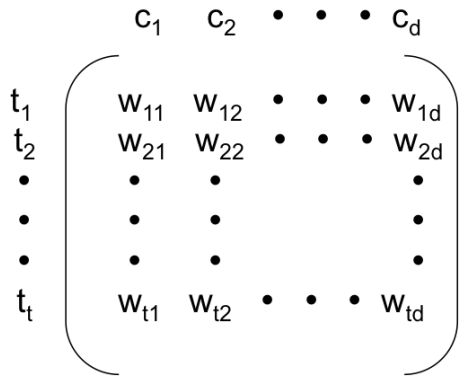


Figure 3.3: Matrice Descripteurs-Classes

Dans la suite de ce manuscrit, nous proposons d'utiliser une matrice Descripteurs-Classes comme dans [49, 190]. Cette représentation permet de construire des modèles simples et robustes de classification de documents. Elle permet aussi de mettre en évidence les descripteurs les plus intéressants pour chaque classe rendant ces modèles compréhensibles lors de la phase d'apprentissage. Enfin, les modèles basés sur une matrice descripteurs-classes permettent une meilleure compréhension des décisions prises par le système lors de la phase de classification.

Un corpus composé de c classes où le dictionnaire est composé de t descripteurs peut être représenté par une matrice $\mathbf{X}$ $c \times t$ avec c lignes et t colonnes. $\mathbf{X}$ est une matrice Descripteurs-Classes et $w_{i,j}$ est le poids du $j^{ème}$ descripteur de la $i^{ème}$ classe.

De nombreuses méthodes ont été proposées dans la littérature pour définir le poids $w_{i,j}$ d'un descripteur.
Elles peuvent généralement être divisées en deux groupes [81] :

- Les méthodes de pondérations supervisées dans lesquelles on va utiliser les informations relatives à l'appartenance de la classe.

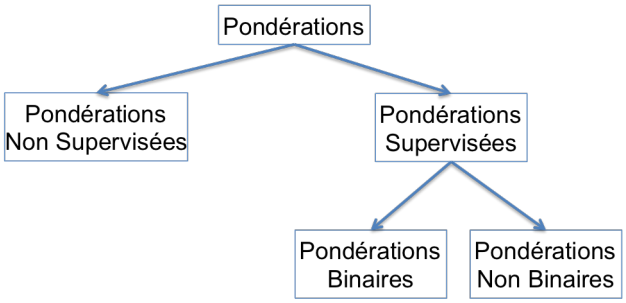


Figure 3.4: Les différents types de pondérations

- Les méthodes de pondérations non supervisées qui ne tiennent pas compte de cette information.

Il est ensuite possible de diviser l'ensemble des pondérations dites supervisées en deux groupes, les pondérations supervisées binaires et les pondérations supervisées non binaires. Les différentes catégories de pondérations, que nous allons expliciter ci-après, sont résumées dans la figure 3.4.

3.2 Les méthodes de pondérations non supervisées

La pondération du descripteur est une étape importante pour améliorer l'efficacité des classifieurs [86]. L'idée de la pondération est de quantifier le poids d'un descripteur en fonction de son importance afin de le différencier des autres. Intuitivement, il est assez simple d'imaginer que si un même descripteur apparait dans une classe A mais pas dans une classe B, il ne peut avoir le même poids pour A et B dans la représentation vectorielle. Avec une pondération booléenne, le poids d'un descripteur vaut 1 s'il apparaît, 0 sinon. De même si un même descripteur apparaissait dans deux classes, cela ne veut pas signifier qu'il ait une importance similaire. Si on utilise la fréquence du descripteur, le poids d'un descripteur vaudra le nombre d'occurrences du descripteur dans la classe. Définir le poids des descripteurs implique deux phases : (1) classer les descripteurs selon leur représentation (savoir quel descripteur est plus représentatif que l'autre) (2) ajuster les poids pour mettre en avant les descripteurs les plus discriminants et limiter le poids des descripteurs les moins importants pour la classification.

La fréquence et la pondération booléenne, bien qu'étant des pondérations assez intuitives, ne sont pas forcément les plus adaptées. Dans un contexte où l'objectif final est la comparaison de vecteurs, l'hypothèse souvent retenue est que deux vecteurs partageant des descripteurs rares est plus discriminant que deux vecteurs partageant des descripteurs fréquents [171]. Cela rejoint les hypothèses retenues en théorie de l'information qui vaut qu'un évènement surprenant ait une importance

plus grande qu'un évènement attendu [147].
Les auteurs dans [192] ont émis les 3 hypothèses suivantes :

1. Les descripteurs rares ne sont pas moins importants que les descripteurs fréquents.
2. Les descripteurs revenant plusieurs fois dans un document ne sont pas moins importants que ceux revenant une seule fois.
3. Pour une même quantité de descripteurs candidats, les documents les plus longs ne sont pas plus importants.

La pondération la plus utilisée pour représenter cette idée est le *TF.IDF* et ses dérivés [65, 137], pondérations issues du monde de la recherche d'information. Cette variante prend en compte la longueur des documents dans le calcul de discriminance.

Le principe du $TF.IDF$ est de donner un poids plus important aux descripteurs les plus spécifiques d'un document [138]. Elle 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 descripteur dans un document et représente le poids du descripteur au sein du document, aussi appelé poids intra-document. Pour un document d_j et un descripteur t_i, la fréquence du descripteur 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 descripteur t_i apparait dans le document d_j et où le dénominateur correspond au nombre total de descripteurs du document d_j. La fréquence inverse de document (IDF) mesure l'importance du descripteur dans le corpus. L'objectif est de donner un poids plus important aux descripteurs 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 descripteur 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 descripteur t_i.

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

Une des variantes les plus utilisées du $TF.IDF$, est la mesure *Okapi-*BM_{25} [129, 64, 130].

Bien que ces pondérations soient la plupart du temps utilisées dans la littérature dans une matrice Descripteurs-Documents, il est tout à fait possible de les généraliser dans une matrice Descripteurs-Classes en considérant simplement l'ensemble des documents d'une classe comme un seul document.

Les 4 pondérations abordées jusqu'à présent (booléen, fréquence, $TF.IDF$ et $Okapi\text{-}BM_{25}$) font donc partie des méthodes de pondérations non supervisées. Les approches de pondérations supervisées utilisant une information complémentaire liée à la classe, il serait logique qu'elles soient plus précises et donc plus pertinentes.

3.3 Les méthodes de pondérations supervisées

Il est possible de catégoriser les méthodes de pondérations supervisées en deux sous-familles. La première regroupe les méthodes qui considèrent le problème de pondération comme un problème de classification binaire (2 classes, la classe positive et les autres classes regroupées dans une classe globale, la classe négative), la seconde va regrouper les méthodes qui considèrent l'ensemble des classes du corpus comme autant de classes indépendantes.

3.3.1 Les méthodes binaires

Plusieurs approches ont été utilisées pour prendre en compte les connaissances apportées par l'appartenance ou non aux classes.
La première consiste à utiliser des métriques de sélection de descripteurs basées sur des mesures statistiques ou issues de la théorie de l'information comme le χ^2 [35, 33], l'*information gain* [33], le *gain ratio* [33, 106] ou encore l'*odds ratio* [79]. Ces mesures étant performantes pour la sélection de descripteurs, elles le sont naturellement pour la définition d'un poids approprié pour une classe donnée. Par exemple, les auteurs dans [33] proposent de capitaliser sur la phase de sélection des descripteurs pour pondérer les descripteurs. Au lieu d'utiliser le score pour déterminer s'il faut ou pas garder le descripteur dans l'espace de recherche, on utilise le score comme pondération. Ils évaluent des mesures comme le $TF.\chi^2$, *TF.information gain* et *TF.gain ratio* pour augmenter le poids des descripteurs discriminants qui sera plus faible si le descripteur est indépendant de la classe et élevé si le descripteur est hautement dépendant de la classe. [106] utilise une méthode basée sur le *gain ratio* et conclut que ce type de pondération est pertinent dans un contexte de résumé.

Plus adaptée pour la classification binaire, la méthode de pondération $TF.rf$ (Relevance Frequency) est présentée dans [79]. L'idée générale est de considérer le nombre de documents de la classe positive et négative qui contiennent le descripteur x. L'intuition du $TF.rf$ est de donner un poids plus important aux descripteurs qui apparaissent le plus fréquemment dans la classe positive [80].

Le $TF.rf$ est asymétrique, car il favorise la classe positive uniquement donc le choix de la classe positive impacte sur la pondération, deuxièmement le $TF.rf$ ne pénalise pas les descripteurs apparaissant aussi fréquemment dans l'ensemble des classes.
Les auteurs dans [97] introduisent le $TF.\delta Idf$ en considérant le nombre de documents dans la catégorie positive (resp. négative) qui contiennent le descripteur. Dans [111], les auteurs utilisent un $TF.KL$ basé sur la divergence de Kullback-Leibler qui exploite la distribution de descripteurs [3].Une approche similaire est utilisée dans [161].

Dans [79, 81], les auteurs utilisent 3 facteurs pour déterminer le poids d'un descripteur :

1. *Term-Frequency Factor* permet de considérer la représentation d'un descripteur au sein de la classe (ce que nous appellerons la représentation intra-classe) ;
2. *Collection-Frequency Factor* prend en considération le comportement de ce descripteur dans l'ensemble de la collection (ce que nous appellerons la représentation inter-classes) ;
3. *Normalization Factor* permet de normaliser les poids selon la taille des documents.

Concernant, le *Term-Frequency Factor*, les auteurs évaluent les mesures suivantes :

- Fréquence du terme (TF), le nombre de fois où le descripteur apparait dans le document ;
- Fréquence normalisée du terme, nombre de fois où le descripteur apparait dans le document divisé par le nombre de descripteurs du document ;
- Logarithme du TF : $\log(1 + tf)$ [86] ;
- L'inverse du TF (ITF) : $1 - \frac{r}{r+tf}$ avec r = 1 généralement [21] ;
- Booléen, utilisé dans tous les algorithmes d'apprentissage supervisé, et plus spécialement avec les approches basées sur la loi de Bernoulli dans lesquelles il n'est pas possible d'utiliser la fréquence.

L'étude conclue qu'il n'y a pas de différence fondamentale entre les 4 approches basées sur la fréquence comme il peut y avoir avec l'approche booléenne.

Concernant, l'évaluation du *Collection-Frequency Factor*, les auteurs citent les mesures suivantes :

- Booléen (1.0) : utilise simplement le *Term-Frequency Factor* ;
- IDF classique : multiplie le poids intra-classe par un IDF ;

- IDF probabiliste : multiplie le poids intra-classe par un IDF probabiliste [184] ;
- χ^2 : multiplie le poids intra-classe par le χ^2 ;
- *information gain* : multiplie le poids intra-classe par l'*information gain* ;
- *gain ratio* : multiplie le poids intra-classe par le *gain ratio* ;
- *odds ratio* : multiplie le poids intra-classe par le *odds ratio*.

Les auteurs concluent que les poids inter-classes basés sur l'IDF ont tendance à diminuer le pouvoir de discrimination des descripteurs par rapport à la classe et qu'il n'y a pas de différence significative entre les différentes transformations de l'IDF.

Enfin *Normalization Factor* est utilisé pour éliminer l'impact de la taille des documents, les auteurs utilisent une normalisation cosinus pour que les valeurs des poids soient comprises entre 0 et 1.

Plutôt que de considérer l'ensemble des autres classes du corpus comme une seule méta-classe, d'autres approches vont considérer l'ensemble des classes séparément.

3.3.2 Les méthodes non binaires

Les approches non binaires reposent sur le postulat qu'il existe une différence entre un descripteur qui apparait dans un grand nombre de classes et un descripteur qui apparait dans un nombre restreint de classes. Un descripteur significatif est un descripteur qui [89] :
- Doit apparaitre fréquemment dans une classe donnée ;
- Doit apparaitre dans peu de documents ;
- Doit être distribué de façon équivalente dans l'ensemble de la collection ;
- Doit être distribué de façon inégale entre les classes ;
- Ne doit pas être distribué de façon différente dans les documents au sein d'une même classe.

Pour mesurer cela, les auteurs dans [89] étendent les travaux initiés dans [88] et introduisent les notions inter-classes, intra-classe et in-collection (dans tout le corpus) à travers trois types de distribution :
- Le facteur inter-classes *Inter-class Stantard Deviation* qui sera élevé pour les descripteurs apparaissant dans toutes les classes, mais à des fréquences inégales. Il est indépendant des classes.
- Le facteur intra-classe *Class Stantard Deviation* qui sera faible si (1) le nombre d'occurrences du descripteur est similaire dans l'ensemble des documents de la classe ou (2) le descripteur apparait rarement dans la classe. Il est propre à chaque classe.
- Le facteur in-collection *Standard Deviation* qui sera faible si (1) le nombre d'occurrences du descripteur est similaire dans l'ensemble des documents du corpus

ou (2) le descripteur apparait rarement dans le corpus. Il est indépendant des classes.

Le poids final d'un descripteur est calculé par multiplication des trois éléments et des coefficients sont utilisés pour modifier l'impact de ces facteurs.

Des pondérations dérivées du $TF.IDF$ ou de *Okapi-*BM_{25} ont aussi été utilisées. Ces pondérations sont des analogies transposées au niveau des documents et des classes (un DF, *Document Frequency*, à la place du TF, un ICF, *Inverse Class Frequency*, à la place d'un IDF). Elles sont utilisées en complément [87] ou en remplacement de l'IDF [49, 190].

3.4 Discussion

Au regard des différents travaux, il apparait qu'il n'existe pas d'approche ou de pondérations qui feraient consensus. Les auteurs dans [33] concluent que le $TF.\chi^2$ et *TF.gain ratio* donnent de meilleurs résultats que *TF.information gain* et qu'en général, les résultats des macro moyennes sont supérieurs au $TF.IDF$ mais pas les micro moyennes. Les auteurs dans [35] montrent que le $TF.\chi^2$ est plus performant que le $TF.IDF$ avec une approche de classification de documents basée sur SVM alors que [33] montre l'inverse ($TF.IDF$ plus performant que $TF.\chi^2$ et que les autres approches pourtant plus complexes).

Dans [81], les auteurs se demandent si les approches de pondérations supervisées donnent de meilleurs résultats que les approches de pondérations non supervisées dans un contexte de classification de documents. Des comparaisons partielles ont été réalisées dans [33] et [156] et n'ont pas permis de tirer de conclusions franches et définitives. De même les conclusions de [35] sont contraires à celles de [79]. Dans [156], les auteurs estiment la proportion de documents contenant le descripteur t_k avec une estimation fondée sur la méthode Wilson [179]. Si les expérimentations donnent de meilleurs résultats avec cette approche plutôt qu'avec des pondérations $TF.IDF$ et *gain ratio*, elles ne montrent pas, en revanche, que les méthodes de pondérations supervisées donnent de meilleurs résultats que les approches de pondérations non supervisées. Dans [51], les auteurs proposent de combiner méthode d'optimisation et méthode des plus proches voisins pour ajuster le poids. Le principal inconvénient de cette approche est le temps nécessaire pour affiner le poids.

Dans [81], les auteurs arrivent à la conclusion que les méthodes de pondérations (qu'elles soient supervisées ou non-supervisées) donneront des résultats différents sur 2 corpus différents ou avec 2 algorithmes différents.

La seconde question que posent les auteurs dans [81] est de savoir si les différences de résultats entre approches de pondérations supervisées et non supervisées sont en lien avec les algorithmes de classification utilisée sans parvenir à une conclusion franche puisqu'il semblerait que les performances varient en fonction des pondérations et non en fonction du type de pondération, mais que les conclusions seront de toute façon différentes en fonction du corpus.

Comme nous venons de le voir, la prise en compte d'une pondération au niveau d'une classe peut reposer sur deux éléments, l'un est toujours présent, il s'agit de la représentation du descripteur au sein de la classe étudiée. Nous parlerons de pondération intra-classe dans la suite du manuscrit. Le second élément est utilisé dans les pondérations dites supervisées en utilisant en complément du poids intra-classe, la représentation des descripteurs au sein des autres classes. Nous parlerons de pondération inter-classes dans la suite du manuscrit. Le poids global est obtenu par la multiplication des deux pondérations intra-classe et inter-classes.

Pour le calcul du poids intra-classe, il est possible de regrouper les différentes pondérations en deux catégories, celles basées sur la fréquence des descripteurs (Term-Frequency) et celles basées sur la fréquence de documents (Document-Frequency). Pour chacune, il existe des dérivés et des transformations par application de fonctions mathématiques ou par addition de variables numériques, mais la base est invariablement distincte, soit on considère la fréquence des documents, soit on considère la fréquence des descripteurs. Ainsi se focaliser sur l'un se fera toujours au détriment de l'autre et impliquera de privilégier l'une des représentations au détriment de l'autre. Prenons un exemple, considérons une classe Canidae qui contiendrait 3 documents composés des descripteurs "chien", "mammifère" et "compagnie" avec une fréquence variable comme illustrée dans le tableau 3.1.

Table 3.1: Classe Canidae

Classe	Document	chien	mammifère	compagnie
Canidae	*can_1*	3	6	1
	can_2	2	0	1
	can_3	1	0	1

Si nous utilisons une approche basée sur le descripteur, nous allons considérer que les descripteurs "chien" et "mammifère" apparaissent aussi fréquemment l'un que l'autre (6 fois) et plus fréquemment que le descripteur "compagnie" (3 fois). À l'inverse si nous utilisons une approche basée sur les documents, nous allons considérer que les descripteurs "chien" et "compagnie" apparaissent aussi fréquemment au sein des documents l'un que l'autre (3 documents) et plus fréquemment que le descripteur "mammifère" (1 seul document). Dans les deux cas, nous nous apercevons que nous allons considérer que le descripteur "chien" qui est présent fréquemment dans de nombreux documents est aussi représentatif qu'un autre descripteur qui apparait soit aussi fréquemment, mais dans peu de documents

(le descripteur "mammifère"), soit dans autant de documents, mais avec une fréquence bien moindre (le descripteur "compagnie"). Nous présentons dans la Section 4.2 des mesures permettant de combiner les deux approches afin de prendre en compte simultanément la fréquence du descripteur et la fréquence du document.

Concernant les approches supervisées, nous observons aussi cette distinction en deux catégories différentes lors du calcul du poids inter-classe. Les approches binaires vont se concentrer sur les documents alors que les approches non-binaires vont se focaliser sur les classes. Nous pourrions nous attendre à ce que les approches non-binaires soient plus performantes que les approches binaires dans la mesure où les informations complémentaires sont utilisées de façon plus fines et précises. En effet, dans une approche binaire, nous allons considérer l'ensemble des autres classes comme étant une seule et même classe négative alors que dans une approche non-binaire nous allons considérer chaque classe comme autant de classes négatives. Intuitivement, nous pouvons supposer qu'un descripteur apparaissant de façon régulière dans l'ensemble des classes négatives soit moins représentatif qu'un descripteur apparaissant globalement plus fréquemment, mais dans une seule classe négative. Néanmoins, en perdant la notion de documents pour ne garder que celle de classe, nous perdons la vision de la représentation au sein de chaque classe. Intuitivement, nous pourrions aussi supposer qu'un descripteur apparaissant de façon régulière dans l'ensemble des documents d'une seule classe négative soit moins représentatif qu'un descripteur apparaissant globalement dans moins de documents, mais dans l'ensemble des classes négatives.

Si nous reprenons notre exemple illustratif. Considérons que nos trois descripteurs ("chien", "mammifère" et "compagnie") apparaissent aussi au sein de documents appartenant à des classes différentes que la classe Canidae, par exemple Felidae et Bovidae. La fréquence des descripteurs est illustrée dans le tableau 3.2. Lors de la pondération inter-classe, nous cherchons à valoriser les descripteurs qui sont rares à l'extérieur de la classe étudiée (Canidae dans notre exemple). Pour rappel, plus ils sont rares dans le reste des classes du corpus et plus ils seront spécifiques (et donc pertinents) pour la classe étudiée.

Table 3.2: Classes Felidae et Bovidae

Classe	Document	chien	mammifère	compagnie
Felidae	*fel_1*	1	1	1
	fel_2	1	0	1
	fel_3	0	0	1
	fel_4	0	0	1
Bovidae	*bov_1*	0	1	0

Si nous utilisons une approche basée sur le document, nous allons considérer que les descripteurs "chien" et "mammifère" apparaissent aussi peu fréquemment l'un que l'autre (dans 2 documents) et moins fréquemment que le descripteur "compagnie" (4 documents). À l'inverse si nous utilisons une approche basée sur la classe,

nous allons considérer que les descripteurs "chien" et "compagnie" apparaissent dans autant de classe (1 classe) et dans moins de classe que le descripteur "mammifère" (qui apparait dans les 2 classes). Là aussi, dans les deux cas, nous allons considérer que le descripteur "chien" qui est présent peu fréquemment et dans peu de documents à l'extérieur de la classe Canidae est aussi représentatif qu'un autre descripteur qui apparait soit aussi fréquemment, mais dans plus de classes (le descripteur "mammifère"), soit dans autant de classes, mais dans un nombre bien plus important de documents (le descripteur "compagnie"). Nous présentons dans la Section 4.4 et démontrons expérimentalement dans le Chapitre 5 qu'une mesure combinant linéairement les 2 approches peut s'avérer pertinente. De plus, les mesures proposées se focalisent soit sur les documents, soit sur les classes. Nous pensons aussi que le descripteur peut être porteur d'une information intéressante en complément pour différencier des descripteurs apparaissant dans un nombre similaire de classes et dans un nombre similaire de documents, mais à des fréquences différentes.

Chapitre

4 Vers l'intégration de nouvelles pondérations

Les différentes pondérations utilisées dans la littérature sont fondées sur des hypothèses qui les rendent performantes pour un type de corpus donné et notre objectif est de définir le poids d'un descripteur donné pour une classe donnée qui soit performant dans un contexte de petits volumes.
Considérons le corpus exemple présenté dans le tableau 4.1. Il est composé de 3 classes (Bovidae, Canidae et Felidae), chacune d'elles étant composée de 3 documents. Pour la présentation des mesures, nous considérons un corpus exemple équilibré (nous discutons du déséquilibre entre classes dans le Chapitre 10).

Table 4.1: Corpus illustratif

Classe	Documents	Descripteurs
Bovidae	bov_1	animal-compagnie-bœuf-génisse-vachette-yack-yack-yack
	bov_2	animal-bœuf-taureau
	bov_3	animal-yack
Canidae	can_1	animal-compagnie-chien-chienne-chiot-toutou
	can_2	animal-compagnie-chiot-chiot-toutou-toutou
	can_3	animal-compagnie-milou-medor-rantanplan-pif
Felidae	fel_1	animal-chat-compagnie-matou-matou-minet
	fel_2	animal-compagnie-matou-minou-minou-minou
	fel_3	animal-compagnie-matou-minou-minou-minou

Dans le tableau 4.1, le document *bov_1* par exemple fait référence au premier document de la classe *Bovidae* et contient 8 descripteurs, 3 fois le descripteur "yack" et 1 fois les descripteurs "animal", "compagnie", "bœuf", "génisse" et "vachette". Le corpus exemple est donc composé de 3 classes, de 9 documents, de 49 descripteurs et de 21 descripteurs uniques (qui composent le dictionnaire).

Nous souhaitons mesurer l'importance de chaque descripteur par rapport à une classe donnée que nous utiliserons lors de la classification pour déterminer la classe

d'un document non labélisé. Pour cela nous cherchons à pondérer chacun des descripteurs du dictionnaire pour chacune des classes. Par exemple, nous cherchons le poids du descripteur "animal" pour la classe Bovidae, mais aussi le poids du même descripteur "animal" pour la classe Canidae et la classe Felidae, les trois poids pouvant potentiellement être différents. Notre proposition repose sur une combinaison entre le poids intra-classe du descripteur (ce que représente le descripteur "animal" pour la classe Bovidae) et le poids inter-classe du descripteur (ce que représente le descripteur "animal" dans les autres classes).

Après avoir présenté l'intuition de notre proposition (Section 4.1), nous présentons une méthode pour évaluer le poids intra-classe (Section 4.2) et le poids inter-classes (Section 4.3). Puis nous définissons une mesure globale reposant sur une combinaison linéaire des poids intra-classe et inter-classes (Section 4.4) avant de conclure ce chapitre par une discussion sur les avantages de notre proposition (Section 4.5).

4.1 Proposition de nouvelles pondérations

Au regard du schéma illustrant tout corpus, notre intuition de base est que la pondération d'un descripteur pour une classe donnée dépend de cinq facteurs :

1. Le descripteur revient-il fréquemment ou peu fréquemment dans la classe ?
2. Le descripteur apparait-il dans beaucoup ou peu de documents de la classe ?
3. Le descripteur apparait-il dans beaucoup ou peu d'autres classes ?
4. Le descripteur apparait-il dans beaucoup ou peu de documents au sein des autres classes ?
5. Le descripteur revient-il fréquemment ou peu fréquemment dans les autres classes ?

Ces cinq facteurs nous permettent d'obtenir une vision exhaustive de la représentation d'un descripteur donné pour une classe donnée, quelque soit la composition du corpus. En effet, comme écrit précédemment, deux descripteurs peuvent être utilisés un même nombre de fois, mais dans un nombre de documents différents ou au contraire apparaitre dans un même nombre de documents, mais un nombre différent de fois. De la même façon, ils peuvent être utilisés un même nombre de fois dans un nombre de classes identiques, mais dans un nombre de documents différents (ou toute autre combinaison possible de ces trois facteurs).

Nous pouvons observer cinq éléments qui permettent de mesurer les cinq facteurs cités précédemment.

1. Les documents de la classe étudiée ;

2. Les descripteurs de la classe étudiée ;
3. Les autres classes ;
4. Les documents des autres classes ;
5. Les descripteurs des autres classes.

Il est possible de regrouper ces éléments en deux catégories, selon qu'ils concernent la classe étudiée (items 1 et 2) ou qu'ils concernent les autres classes du corpus (items 3, 4, 5). Ces deux catégories correspondent aux notions d'**intra-classe** (au sein de la classe) et d'**inter-classes** (au sein des autres classes) comme résumé dans le tableau 4.2.

Table 4.2: Tableau récapitulatif des éléments

Éléments	**Périmètre**	**Catégories**
documents	classe étudiée	intra-classe
descripteurs	classe étudiée	intra-classe
classes	autres classes	inter-classes
documents	autres classes	inter-classes
descripteurs	autres classes	inter-classes

Dans la section suivante, nous nous intéressons, pour commencer, à la définition d'un poids intra-classe.

4.2 La représentation intra-classe

Nous avons précisé dans la section précédente que nous pouvions utiliser deux éléments pour calculer le poids intra-classe d'un descripteur : les documents et les descripteurs.
Dans cette section, nous présentons tout d'abord une pondération intra-classe fondée sur le document que nous appelons *intra-classe*^{DF} (Section 4.2.1) puis une pondération intra-classe fondée sur le descripteur que nous appelons poids *intra-classe*^{TF} (Section 4.2.2). Pour illustrer nos propos, nous nous appuierons sur le corpus exemple présenté en introduction de chapitre.

4.2.1 *Intra-classe*^{DF}

Dans un premier temps, nous proposons une méthode de pondération fondée sur la fréquence de documents du descripteur dans la classe comme décrite dans [49]. Nous appelons cette pondération *intra-classe*^{DF} puisqu'elle se base sur la fréquence de documents (DF : *Document Frequency*) et nous la formalisons dans la Définition n°1.

Définition 1 ($Intra\text{-}classe_{ij}^{DF}$)

$Intra\text{-}classe_{ij}^{Df}$: Pondération mesurant la représentativité du descripteur t_i pour la classe C_j fondée sur la fréquence des documents de la classe C_j

Avec : $intra\text{-}classe_{ij}^{DF} = \frac{DF_{ti}^{j}}{|d_j|}$

Où :
- DF_{ti}^{j} correspond au nombre de documents de la classe C_j qui contiennent le descripteur t_i et
- $|d_j|$ correspond au nombre total de documents de la classe C_j

Par exemple, nous calculons les poids $intra\text{-}classe^{DF}$ pour les descripteurs "minou" et "compagnie" de la classe Felidae (Exemple n°1).

Exemple 1 ($intra\text{-}classe_{minou,Felidae}^{DF}$ et $intra\text{-}classe_{compagnie,Felidae}^{DF}$)

$$intra\text{-}classe_{minou,Felidae}^{DF} = \frac{DF_{minou}^{Felidae}}{|d_{Felidae}|} = \frac{|\{fel\_2, fel\_3\}|}{|\{fel\_1, fel\_2, fel\_3\}|} = \frac{2}{3} = 0.67$$

$$intra\text{-}classe_{compagnie,Felidae}^{DF} = \frac{DF_{compagnie}^{Felidae}}{|d_{Felidae}|} = \frac{|\{fel\_1, \{fel\_2, fel\_3\}|}{|\{fel\_1, fel\_2, fel\_3\}|} = \frac{3}{3} = 1$$

Dans notre exemple, "compagnie" est considéré comme plus représentatif que "minou" ($1 > 0.67$). Avec cette pondération, nous considérons que le descripteur le plus représentatif d'une classe n'est pas nécessairement le descripteur le plus fréquemment utilisé dans la classe, mais celui utilisé dans le plus grand nombre de documents de la classe. Une telle mesure peut se révéler particulièrement pertinente pour le traitement de documents de longueurs déséquilibrées puisque dans ce cas, les descripteurs présents dans les documents de plus grandes tailles auront un impact similaire aux descripteurs présents dans les plus petits documents.

Néanmoins la pondération $lntra\text{-}classe^{DF}$ a pour effet de lisser la réalité des descripteurs au sein de chaque document et d'occulter une information pourtant importante. Dans notre exemple, le descripteur "minou" bien qu'apparaissant deux

fois plus souvent va être considéré comme moins représentatif de la classe Felidae que le descripteur "compagnie", car il apparait dans un nombre de documents moindre. De plus, avec les classes composées d'un faible nombre de documents, 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 descripteur plutôt que sur le document dans la section suivante.

4.2.2 *Intra-classe*^{TF}

Avec cette pondération, nous considérons la fréquence du descripteur plutôt que la fréquence du document et nous appelons cette pondération *intra-classe*^{TF} puisqu'elle est fondée sur la fréquence du terme (TF : *Term Frequency*) et nous la formalisons dans le Définition n°2.

Définition 2 (*Intra-classe*$^{TF}_{ij}$)

Intra-classe$^{Tf}_{ij}$: Pondération mesurant la représentativité du descripteur t_i pour la classe C_j fondée sur la fréquence des termes de la classe C_j

Avec : *intra-classe*$^{Tf}_{ij} = \frac{TF^j_{ti}}{|n_j|}$

Où :
- TF^j_{ti} correspond au nombre d'occurrences du descripteur t_i de la classe C_j et
- $|n_j|$ correspond au nombre total de descripteurs de la classe C_j

Si nous reprenons notre exemple, nous calculons les poids *intra-classe*^{TF} pour les descripteurs "minou" et "compagnie" (Exemple n°2) de la classe Felidae.

Exemple 2 (*intra-classe*$^{TF}_{minou,Felidae}$ et *intra-classe*$^{TF}_{compagnie,Felidae}$)

$$intra\text{-}classe^{TF}_{minou,Felidae} = \frac{TF^{Felidae}_{minou}}{|n_{Felidae}|} = \frac{6}{18} = 0.33$$

$$intra\text{-}classe^{TF}_{compagnie,Felidae} = \frac{TF^{Felidae}_{compagnie}}{|n_{Felidae}|} = \frac{3}{18} = 0.17$$

Dans notre exemple, selon la pondération *intra-classe*^{TF}, "minou" est considéré comme plus représentatif que "compagnie" ($0.33 > 0.17$) pour la classe Felidae. Avec cette pondération, nous considérons que le descripteur le plus représentatif d'une classe est le descripteur le plus fréquemment utilisé dans la classe, indépendamment du nombre de documents qui contient le descripteur. Une telle mesure peut se révéler particulièrement inadaptée pour le traitement de documents de longueurs déséquilibrées puisque, dans ce cas, les descripteurs présents dans les documents de plus grandes tailles auront un impact disproportionné par rapport aux descripteurs présents dans les plus petits documents.

Dans cette section, nous venons de définir deux mesures *intra-classe*^{DF} et *intra-classe*^{TF} permettant de mesurer la représentativité d'un descripteur au sein d'une classe. Ces deux mesures prises indépendamment affichent des limites selon les corpus étudiés. Nous verrons ultérieurement dans la section 4.4 comment les cumuler afin de gommer leurs biais respectifs.

Nous avons établi à la Section 4.1, qu'un poids pertinent pour un descripteur donné d'une classe donnée était composé à la fois de sa représentation intra-classe, que nous venons d'étudier, mais également de sa représentation inter-classes. Nous étudions cette représentation inter-classes dans la section suivante.

4.3 La représentation inter-classes

Pour évaluer le poids inter-classes d'un descripteur donné pour une classe donnée, la pondération inter-classes va se focaliser sur le comportement du descripteur dans les autres classes que la classe étudiée. Avec la pondération inter-classes, nous cherchons à valoriser non pas les descripteurs qui apparaissent fréquemment dans le reste du corpus, mais les descripteurs qui sont rares : l'idée sous-jacente étant que plus ils sont rares dans le reste du corpus et plus ils seront spécifiques et donc discriminants pour la classe étudiée.
Dans la section 4.1, nous sommes parvenus à la conclusion que le poids d'un descripteur pouvait dépendre de cinq facteurs, dont deux permettent de mesurer le poids intra-classe et les trois autres le poids inter-classes (tableau 4.3).

Table 4.3: Tableau récapitulatif des pondérations

Eléments	Périmètre	Catégories	Pondérations	Référence
documents	classe étudiée	intra-classe	***intra-classe***^{DF}	Section 4.2.1
descripteurs	classe étudiée	intra-classe	***intra-classe***^{TF}	Section 4.2.2
classes	autres classes	inter-classes		
documents	autres classes	inter-classes		
descripteurs	autres classes	inter-classes		

Nous présentons dans cette section trois pondérations inter-classes en commençant par une pondération fondée sur les classes.

4.3.1 *Inter-classes*^{ICF}

Nous proposons une première méthode que nous nommons *inter-classes*^{ICF} (Définition n°3) et qui s'appuie sur l'inverse de la fréquence de classe (ICF : *Inverse Class Frequency*).

Définition 3 (*Inter-classes*$_{ij}^{ICF}$)

Inter-classes$_{ij}^{ICF}$: Pondération mesurant l'inverse de la représentativité du descripteur t_i pour les classes autres que C_j en se basant sur les classes.

Avec : $inter\text{-}classes_{ij}^{ICF} = \log_2(\frac{|C|}{|C_{ti}|})$

Où :
- $|C|$ correspond au nombre total de classes du corpus et
- $|C_{ti}|$ correspond au nombre de classes du corpus qui contient le descripteur t_i

La fonction logarithme est utilisée pour diminuer les écarts et donc l'influence entre les poids inter-classes comme cela est fait notamment dans les approches $TF\text{-}IDF$. Cette méthode, fondée sur le nombre de classes du corpus qui contiennent le descripteur, a été utilisée dans [49].

Pour illustrer, nous reprenons notre exemple (que nous rappelons pour des raisons de lisibilité dans le tableau 4.4) et que nous évaluons, toujours pour la classe Felidae, la pondération *inter-classes*^{ICF} des descripteurs "minou" et "compagnie" (Exemple n°3).

Table 4.4: Corpus illustratif

Classe	Documents	Descripteurs
Bovidae	bov_1	animal-compagnie-bœuf-génisse-vachette-yack-yack-yack
	bov_2	animal-bœuf-taureau
	bov_3	animal-yack
Canidae	can_1	animal-compagnie-chien-chienne-chiot-toutou
	can_2	animal-compagnie-chiot-chiot-toutou-toutou
	can_3	animal-compagnie-milou-medor-rantanplan-pif
Felidae	fel_1	animal-chat-compagnie-matou-matou-minet
	fel_2	animal-compagnie-matou-minou-minou-minou
	fel_3	animal-compagnie-matou-minou-minou-minou

Exemple 3 ($inter\text{-}classes^{ICF}_{minou,Felidae}$ et $inter\text{-}classes^{ICF}_{compagnie,Felidae}$)

$$
\begin{aligned}
&inter\text{-}classes^{ICF}_{minou,Felidae} \\
&= \log_2(\tfrac{|C|}{|C_{minou}|}) \\
&= \log_2(\tfrac{|\{Bovidae,Canidae,Felidae\}|}{|\{Felidae\}|}) \\
&= \log_2(\tfrac{3}{1}) \\
&= \log_2(3) \\
&= 1.58
\end{aligned}
$$

$$
\begin{aligned}
&inter\text{-}classes^{ICF}_{compagnie,Felidae} \\
&= \log_2(\tfrac{|C|}{|C_{minou}|}) \\
&= \log_2(\tfrac{|\{Bovidae,Canidae,Felidae\}|}{|\{Bovidae,Canidae,Felidae\}|}) \\
&= \log_2(\tfrac{3}{3}) \\
&= \log_2(1) \\
&= 0
\end{aligned}
$$

Dans notre exemple, le descripteur "compagnie" qui apparait dans toutes les classes sera pondéré avec une importance moindre que le descripteur "minou" qui n'apparait que dans une seule classe ($0 < 1.58$). Le descripteur "minou" est plus spécifique et ainsi porteur de plus d'information pour des tâches de recherche d'information. Le principe est similaire à celui de l'IDF mais transposé à la classe.

Cependant, nous estimons que cette approche, qui consiste à considérer la présence ou l'absence d'un descripteur dans une classe sans prendre en compte la fréquence, est trop restrictive, car il suffit qu'un descripteur soit utilisé une seule fois dans un seul document d'une classe pour le considérer comme représentatif de la classe. Nous pensons par exemple, que l'approche $inter\text{-}classes^{ICF}$ sera trop restrictive dans les cas de figure suivants :

- *Présence de classes sémantiquement proches* : des classes proches sémantiquement vont partager un grand nombre de descripteurs en commun, mais à des fréquences potentiellement différentes.
- *Présence d'un grand nombre de descripteurs par classe* (lié à un grand nombre de documents ou à des documents très longs) : plus le nombre de descripteurs est important, plus la probabilité qu'un descripteur apparaisse au moins une fois par classe est forte.

Suite à ces observations, nous proposons de considérer également l'absence ou la présence d'un descripteur, mais au niveau des documents.

4.3.2 *Inter-classes*^{IDF}

Nous pensons qu'observer et mesurer le comportement d'un descripteur au travers des documents des autres classes peut permettre d'obtenir une vision plus fine.

Nous proposons une deuxième méthode que nous nommons *inter-classes*^{IDF} (Définition n°4) et qui se base sur l'inverse de la fréquence de documents (IDF : *Inverse Document Frequency*).

Définition 4 (*Inter-classes*$_{ij}^{IDF}$)

Inter-classes$_{ij}^{IDF}$: Pondération mesurant l'inverse de la représentativité du descripteur t_i pour les classes autres que C_j en se basant sur les documents.

Avec : $inter\text{-}classes_{ij}^{IDF} = \log_2(\frac{|d|-|d\in C_j|+1}{|d:t_i|-|d:t_i\in C_j|+1})$

Où :
- $|d|$ est le nombre de documents total dans l'ensemble du corpus
- $|d \in C_j|$ est le nombre de documents total de la classe C_j
- $|d : t_i|$ est le nombre de documents dans l'ensemble du corpus qui contiennent le descripteur t_i
- $|d : t_i \in C_j|$ est le nombre de documents de la classe C_j qui contiennent le descripteur t_i
- Ajouter 1 permet d'éviter toute erreur quand t_i est utilisé uniquement dans C_j (quand $|d : t_i| - |d : t_i \in C_j| = 0$)

L'idée générale est de diviser le nombre de documents n'appartenant pas à la classe étudiée (que nous obtenons en soustrayant le nombre de documents de la classe étudiée au nombre total de documents du corpus) par le nombre de documents n'appartenant pas à la classe étudiée qui comporte le descripteur t_i (que nous obtenons en soustrayant le nombre de documents de la classe étudiée qui contiennent t_i au nombre total de documents du corpus qui contiennent t_i).
Nous reprenons notre exemple et nous évaluons, toujours pour la classe Felidae, la pondération *inter-classes*^{IDF} des descripteurs "minou" et "compagnie" (Exemple n°4).

Exemple 4 ($inter\text{-}classes^{IDF}_{minou,Felidae}$ et $inter\text{-}classes^{IDF}_{compagnie,Felidae}$)

$$inter\text{-}classes^{IDF}_{minou,Felidae}$$
$$= \log_2\left(\frac{|d|-|d\in C_{Felidae}|+1}{|d{:}t_{minou}|-|d{:}t_{minou}\in C_{Felidae}|+1}\right)$$
$$= \log_2\left(\frac{|\{bov\_1,bov\_2,bov\_3,can\_1,can\_2,can\_3,fel\_1,fel\_2,fel\_3\}|-|\{fel\_1,fel\_2,fel\_3\}|+1}{|\{fel\_2,fel\_3\}|-|\{fel\_2,fel\_3\}|+1}\right)$$
$$= \log_2\left(\frac{9-3+1}{2-2+1}\right)$$
$$= \log_2\left(\frac{7}{1}\right)$$
$$= 2.81$$

$$inter\text{-}classes^{IDF}_{compagnie,Felidae}$$
$$= \log_2\left(\frac{|d|-|d\in C_{Felidae}|+1}{|d{:}t_{compagnie}|-|d{:}t_{compagnie}\in C_{Felidae}|+1}\right)$$
$$= \log_2\left(\frac{|\{bov\_1,bov\_2,bov\_3,can\_1,can\_2,can\_3,fel\_1,fel\_2,fel\_3\}|-|\{fel\_1,fel\_2,fel\_3\}|+1}{|\{bov\_1,can\_1,can\_2,can\_3,fel\_1,fel\_2,fel\_3\}|-|\{fel\_1,fel\_2,fel\_3\}|+1}\right)$$
$$= \log_2\left(\frac{9-3+1}{7-3+1}\right)$$
$$= \log_2\left(\frac{7}{5}\right)$$
$$= 0.49$$

Bien que corrigeant certains biais introduits par la pondération $inter\text{-}classes^{ICF}$, la pondération $inter\text{-}classes^{IDF}$ possède elle aussi quelques limites. En effet, en occultant la notion de classe pour nous concentrer sur la notion de document, nous considérons de la même façon un descripteur qu'il soit très fréquemment présent dans une seule classe ou repartie de façon uniforme entre les classes. De plus, en cas de corpus fortement déséquilibré (lorsqu'une classe possède bien plus de documents que les autres), les pondérations seront très fortement influencées par la classe majoritaire (celle qui possède le plus de documents). Au même titre que les pondérations intra-classes présentées en Section 4.2, nous nous apercevons qu'il n'existe pas de pondération inter-classes qui soit sans biais indépendamment du corpus. Nous proposons une combinaison des pondérations inter-classes dans la Section 4.4 mais auparavant, nous proposons une troisième pondération inter-classes qui serait fondée sur les descripteurs et que nous présentons dans la section suivante.

4.3.3 *Inter-classes*^{ITF}

Nous pouvons aussi mesurer le comportement d'un descripteur au sein des autres classes du corpus indépendamment des documents ou des classes.

Nous nommons cette troisième méthode $inter\text{-}classes^{ITF}$ (Définition n°5). Celle-ci se base sur l'inverse de la fréquence de termes (ITF : *Inverse Term Frequency*).

Définition 5 ($Inter\text{-}classes_{ij}^{ITF}$)

$Inter\text{-}classes_{ij}^{ITF}$: Pondération mesurant l'inverse de la représentativité du descripteur t_i pour les classes autres que C_j en se basant sur les descripteurs.

Avec : $inter\text{-}classes_{ij}^{ITF} = \log_2(\frac{|t|-|t\in C_j|+1}{|t_i|-|t_i\in C_j|+1})$

Où :
- $|t|$ est le nombre de descripteurs total dans l'ensemble du corpus
- $|t \in C_j|$ est le nombre de descripteurs total de la classe C_j
- $|t_i|$ est le nombre d'occurrences du descripteur t_i dans l'ensemble du corpus
- $|t_i \in C_j|$ est le nombre d'occurrences du descripteur t_i dans la classe C_j
- Ajouter 1 permet d'éviter toute erreur quand t_i est utilisé uniquement dans C_j (quand $|t_i| - |t_i \in C_j| = 0$)

Pour illustrer, nous complétons notre exemple avec la pondération $inter\text{-}classes^{ITF}$ des descripteurs "minou" et "compagnie" (Étude de cas n° 5).

Exemple 5 ($inter\text{-}classes_{minou,Felidae}^{ITF}$ et $inter\text{-}classes_{compagnie,Felidae}^{ITF}$)

$$
\begin{aligned}
&inter\text{-}classes_{minou,Felidae}^{ITF} \\
&= \log_2(\tfrac{|t|-|t\in C_{Felidae}|+1}{|t_{minou}|-|t_{minou}\in C_{Felidae}|+1}) \\
&= \log_2(\tfrac{(8+3+2+6+6+6+6+6+6)-(6+6+6)+1}{(0+0+0+0+0+0+0+3+3)-(0+3+3)+1}) \\
&= \log_2(\tfrac{49-18+1}{6-6+1}) \\
&= \log_2(\tfrac{32}{1}) \\
&= 5
\end{aligned}
$$

$$
\begin{aligned}
&inter\text{-}classes_{compagnie,Felidae}^{ITF} \\
&= \log_2(\tfrac{|t|-|t\in C_{Felidae}|+1}{|t_{compagnie}|-|t_{compagnie}\in C_{Felidae}|+1}) \\
&= \log_2(\tfrac{(8+3+2+6+6+6+6+6+6)-(6+6+6)+1}{(1+0+0+1+1+1+1+1+1)-(1+1+1)+1}) \\
&= \log_2(\tfrac{49-18+1}{7-3+1}) \\
&= \log_2(\tfrac{32}{5}) \\
&= 2.68
\end{aligned}
$$

Bien que rarement utilisée dans la littérature, nous pensons que cette pondération peut être utile lorsque la répartition au sein des classes ou des documents est trop

similaire tout en sachant que ce cas de figure sera assez rare. De plus, en cas de corpus fortement déséquilibré, les pondérations seront très fortement influencées par la classe majoritaire (celle qui possède le plus de mots). Nous résumons les différentes pondérations dans le tableau 4.5.

Table 4.5: Tableau récapitulatif des pondérations

Eléments	Périmètre	Catégories	Pondérations	Référence
documents	classe étudiée	intra-classe	$intra\text{-}classe^{DF}$	Section 4.2.1
descripteurs	classe étudiée	intra-classe	$intra\text{-}classe^{TF}$	Section 4.2.2
classes	autres classes	inter-classes	$inter\text{-}classes^{ICF}$	Section 4.3.1
documents	autres classes	inter-classes	$inter\text{-}classes^{IDF}$	Section 4.3.2
descripteurs	autres classes	inter-classes	$inter\text{-}classes^{ITF}$	Section 4.3.3

Chacune de ces cinq pondérations est accompagnée d'un certain nombre de biais induits par la manière dont elles sont calculées. Pour autant, elles sont toutes porteuses d'une information pertinente représentative de la réalité du corpus et sont toutes complémentaires. C'est pourquoi nous discutons et présentons dans la section suivante une solution pour construire un poids global reposant sur une combinaison des cinq pondérations définies dans ce chapitre.

4.4 Vers une pondération globale

4.4.1 Les cinq composants d'une pondération globale

Tout au long de ce chapitre, nous avons calculé des scores pour deux descripteurs donnés ("minou" et "compagnie") d'une classe donnée (Felidae) à partir de notre corpus exemple pour illustrer les différentes pondérations. Ces pondérations, calculées indépendamment les unes des autres, sont récapitulées dans le tableau 4.6.

Table 4.6: Récapitulatif des pondérations calculées pour "minou" et "compagnie" pour la classe Felidae

		intra-classe		*inter-classes*		
Classe	Descripteur	DF	TF	ICF	IDF	ITF
Felidae	minou	0.67	0.33	1.58	2.81	5
	compagnie	1	0.17	0	0.49	2.68

Or notre objectif est de définir un seul poids pour un descripteur donné d'une classe donnée et non cinq poids différents. Considérons le tableau 4.7 qui reporte l'ensemble des pondérations pour notre corpus.

Table 4.7: Récapitulatif des pondérations intra et inter-classes

		intra-classe		*inter-classes*		
Classe	Descripteur	DF	TF	ICF	IDF	ITF
Bovidae	animal	1	0.23	0	0	2.40
	compagnie	0.33	0.08	0	0	2.40
	bœuf	0.67	0.15	1.58	2.81	5.21
	génisse	0.33	0.08	1.58	2.81	5.21
	taureau	0.33	0.08	1.58	2.81	5.21
	vachette	0.33	0.08	1.58	2.81	5.21
	yack	0.67	0.31	1.58	2.81	5.21
Canidae	animal	1	0.17	0	0	2.19
	chien	0.33	0.06	1.58	2.81	5
	chienne	0.33	0.06	1.58	2.81	5
	chiot	0.67	0.17	1.58	2.81	5
	compagnie	1	0.17	0	0.49	2.68
	medor	0.33	0.06	1.58	2.81	5
	milou	0.33	0.06	1.58	2.81	5
	pif	0.33	0.06	1.58	2.81	5
	rantanplan	0.33	0.06	1.58	2.81	5
	toutou	0.67	0.17	1.58	2.81	5
Felidae	animal	1	0.17	0	0	2.19
	chat	0.33	0.06	1.58	2.81	5
	compagnie	1	0.17	0	0.49	2.68
	matou	1	0.22	1.58	2.81	5
	minet	0.33	0.06	1.58	2.81	5
	minou	0.67	0.33	1.58	2.81	5

Cet exemple nous permet de mettre en évidence un certain nombre de propriétés concernant les pondérations :

1. Les pondérations ont un ordre de grandeur différent :
   - les pondérations intra-classes sont toujours inférieures ou égales à 1 ;
   - les pondérations inter-classes sont potentiellement supérieures à 1.
2. L'ordre de grandeur des variations au sein d'une même pondération est différent selon la pondération :
   - $\textit{intra-classe}^{DF}$ varie généralement plus que $\textit{intra-classe}^{TF}$;
   - $\textit{inter-classes}^{ITF}$ varie généralement plus que $\textit{inter-classes}^{IDF}$, lui-même variant généralement plus que $\textit{inter-classes}^{ICF}$.
3. Les pondérations intra-classes ont un ordre de grandeur différent ;
4. Les pondérations inter-classes ont un ordre de grandeur différent.

Concernant le point 1 (les pondérations ont un ordre de grandeur différent), les cinq pondérations reposent sur la division de deux éléments. Or si le numérateur est inférieur au dénominateur, le résultat est inférieur à 1, dans le cas contraire il est supérieur à 1. Dans le cas des pondérations inter-classes, le numérateur est toujours inférieur ou égal au dénominateur, car nous divisons le nombre de descripteur t_i ou de documents contenant t_i dans la classe par le nombre total de descripteurs ou de documents de la classe. Ainsi, les pondérations intra-classes seront toujours bornées théoriquement entre 0 (si le descripteur n'apparait pas, le numérateur vaut 0) et 1 (si le descripteur apparait dans tous les documents alors le numérateur est égal au dénominateur ou encore s'il n'y avait qu'un seul

descripteur dans la classe[1]). À l'inverse, dans les pondérations inter-classes, le numérateur est toujours supérieur ou égal au dénominateur, car nous divisons le nombre de classes, documents ou descripteurs par le nombre de classes ou de documents contenant t_i ou par le nombre d'occurrences de t_i. Le résultat de la division est donc compris entre 1 (si le descripteur apparait dans toutes les classes, tous les documents ou lorsqu'il n'y a qu'un seul descripteur dans le reste des classes du corpus[2]) et l'infini (ou plus précisément le nombre de classes, documents ou descripteurs). La fonction logarithme appliquée sur des valeurs comprises entre 1 et l'infini donne des résultats compris entre 0 et l'infini. Les pondérations inter-classes sont donc mathématiquement comprises entre 0 et l'infini. Plus précisément, par application de la fonction logarithme binaire, les valeurs inter-classes seront supérieures (resp inférieures) à 1 si le ratio entre le numérateur et le dénominateur est supérieur (resp inférieur) à 2.

Concernant le point 2 (l'ordre de grandeur des variations au sein d'une même pondération est différent), l'explication tient à la relation liant descripteurs, documents et classes. Dans la mesure où il est acquis qu'une classe sans document et un document sans descripteur ne doivent pas exister, une classe sera composée d'un ou plusieurs documents (probablement plusieurs) et un document sera lui même composé d'un ou plusieurs descripteurs (probablement plusieurs également). Il n'est donc pas possible d'avoir plus de classes que de documents, ni plus de documents que de descripteurs et généralement le nombre de descripteurs sera très largement supérieur au nombre de documents, lui-même étant supérieur au nombre de classes. Pour rappel, le corpus exemple est composé de 3 classes, 9 documents et 49 descripteurs.
Ainsi, ajouter une classe supplémentaire aura plus d'impact qu'ajouter un document supplémentaire. De même que l'ajout d'un descripteur supplémentaire aura un impact moindre.

Concernant les points 3 et 4, le nombre de classes, de documents et de descripteurs étant très différent, les pondérations calculées à partir de ceux-ci le seront tout autant à moins de respecter parfaitement les mêmes proportions (ce qui est improbable). De plus, s'il est tout à fait envisageable d'avoir une pondération $intra\text{-}classe^{DF}$ égale à 1 (qui implique que le descripteur soit utilisé dans tous les documents de la classe), il est inenvisageable que la pondération $intra\text{-}classe^{TF}$ de ce même descripteur soit égale à 1 (qui impliquerait qu'il n'y ait qu'un seul descripteur dans la classe). En outre, dans la pratique, le nombre de descripteurs étant largement supérieur au nombre de documents, $intra\text{-}classe^{TF}$ tendra plus fréquemment vers 0 que $intra\text{-}classe^{DF}$.

De même pour le point 4, la probabilité que la pondération $inter\text{-}classes^{ICF}$ soit égale à 0 (impliquant que le descripteur soit utilisé dans toutes les classes) est

1. ce cas-là étant hautement improbable dans la pratique
2. ce cas-là étant lui aussi hautement improbable dans la pratique

beaucoup plus élevée que la probabilité que la pondération $inter\text{-}classes^{IDF}$ soit égale à 0 (impliquant que le descripteur soit utilisé dans tous les documents des autres classes). Enfin, la valeur maximum des poids intra-classes est égale au logarithme du nombre d'éléments (soit de descripteurs, de documents ou de classes). Le nombre de descripteurs étant en général largement supérieur au nombre de documents, lui-même largement supérieur au nombre de classes et la fonction logarithme étant strictement croissante, alors la valeur maximum potentielle de $inter\text{-}classes^{ITF}$ est supérieure à celle de $inter\text{-}classes^{IDF}$, elle même supérieure à celle de $inter\text{-}classes^{ICF}$.

Ces propriétés ont deux conséquences quant au calcul du poids global. Il n'est pas possible de les combiner en l'état, car :

1. la pondération globale sera influencée par les pondérations ayant les plus fortes valeurs et les plus fortes variations
2. la pondération globale sera plus influencée par les poids inter-classes (3 pondérations) que par les poids intra-classes (2 pondérations)

Pour atténuer ces différentes spécificités, nous proposons d'ajouter deux traitements :

1. normaliser indépendamment chaque pondération et les ramener dans un même ordre de grandeur (par exemple entre 0 et 1) afin de limiter les effets induits par les différences de valeurs et de variations.
2. scinder notre mesure globale en deux mesures (que nous qualifierons de mesures locales par opposition à globale), l'une qui regrouperait les pondérations intra-classes, l'autre qui regrouperait les pondérations inter-classes et de considérer les deux mesures locales avec la même importance dans la pondération globale.

Nous étudions dans la Section 4.4.2 la normalisation des pondérations. La mesure globale prenant en compte le second traitement sera détaillée en Section 4.4.3.

4.4.2 La normalisation

Notre objectif est de normaliser les différentes pondérations afin de les rendre comparables. La solution la plus pertinente est d'extraire les maximums des cinq pondérations sur l'ensemble du corpus puis de normaliser chaque pondération par son maximum ($intra\text{-}classe^{DF}$ divisé par le maximum $intra\text{-}classe^{DF}$, $intra\text{-}classe^{TF}$ divisé par le maximum $intra\text{-}classe^{TF}$...). Nous ajoutons le suffixe - $Norm$ au nom des pondérations pour faire référence à la version normalisée (par exemple la définition de la pondération $intra\text{-}classe^{DF} - Norm$ est donnée ci-après) [3].

3. Seule la définition de la pondération $intra\text{-}classe^{DF} - Norm$ est donnée pour exemple dans la Définition 6, les quatre autres pondérations normalisées découlant du même processus.

Définition 6 ($intra\text{-}classe_{ij}^{DF} - Norm$)

$intra\text{-}classe_{ij}^{DF} - Norm$: Pondération $intra\text{-}classe^{DF}$ du descripteur t_i pour la classe C_j obtenue après normalisation.

Avec : $intra\text{-}classe_{ij}^{DF}\text{-}Norm = \frac{intra\text{-}classe_{ij}^{DF}}{\arg\max intra\text{-}classe^{DF}}$

Si la valeur maximum est supérieure à 1 (cas des pondérations inter-classes) alors les valeurs normalisées seront plus faibles que les valeurs réelles et les écarts entre valeurs seront moins importants. À l'inverse si la valeur maximum est inférieure à 1 (cas des pondérations intra-classes) alors les valeurs normalisées seront plus élevées que les valeurs réelles et les écarts entre valeurs seront plus importants. Dans tous les cas, les pondérations normalisées seront toutes comprises entre 0 et 1. Si nous reprenons notre exemple, nous extrayons d'abord les maximums (voir tableau 4.8).

Table 4.8: Extraction des maximums

		intra-classe		*inter-classes*		
Classe	Descripteur	DF	TF	ICF	IDF	ITF
Bovidae	animal	1	0.23	0	0	2.40
	compagnie	0.33	0.08	0	0	2.40
	bœuf	0.67	0.15	1.58	2.81	5.21
	génisse	0.33	0.08	1.58	2.81	5.21
	taureau	0.33	0.08	1.58	2.81	5.21
	vachette	0.33	0.08	1.58	2.81	5.21
	yack	0.67	0.31	1.58	2.81	5.21
Canidae	animal	1	0.17	0	0	2.19
	chien	0.33	0.06	1.58	2.81	5
	chienne	0.33	0.06	1.58	2.81	5
	chiot	0.67	0.17	1.58	2.81	5
	compagnie	1	0.17	0	0.49	2.68
	medor	0.33	0.06	1.58	2.81	5
	milou	0.33	0.06	1.58	2.81	5
	pif	0.33	0.06	1.58	2.81	5
	rantanplan	0.33	0.06	1.58	2.81	5
	toutou	0.67	0.17	1.58	2.81	5
Felidae	animal	1	0.17	0	0	2.19
	chat	0.33	0.06	1.58	2.81	5
	compagnie	1	0.17	0	0.49	2.68
	matou	1	0.22	1.58	2.81	5
	minet	0.33	0.06	1.58	2.81	5
	minou	0.67	0.33	1.58	2.81	5
Maximum		1	0.33	1.58	2.81	5.21

Par exemple le maximum de la pondération $intra\text{-}classe^{DF}$ vaut 1 et est atteint pour les descripteurs :

- "animal" de la classe Bovidae
- "compagnie" et "animal" de la classe Canidae
- "matou", "compagnie" et "animal" de la classe Felidae

Sur la base de notre exemple, le maximum de la pondération $intra\text{-}classe^{TF}$ vaut 0.33 et est atteint uniquement pour le descripteur "minou" de la classe Felidae. Les cinq maximums ainsi obtenus sont :

- $\arg\max intra\text{-}classe^{TF} = 1$
- $\arg\max intra\text{-}classe^{DF} = 0.33$
- $\arg\max inter\text{-}classes^{ICF} = 1.58$
- $\arg\max inter\text{-}classes^{IDF} = 2.81$
- $\arg\max inter\text{-}classes^{ITF} = 5.21$

Puis, nous normalisons les pondérations par les maximums extraits, comme illustré dans l'Exemple n°6.

Exemple 6 (Pondérations normalisées pour "minou" de la classe Felidae)

$$intra\text{-}classe^{DF}_{minou,Felidae}\text{-}Norm = \frac{intra\text{-}classe^{DF}_{minou,Felidae}}{\arg\max intra\text{-}classe^{DF}} = \frac{0.67}{1} = 0.67$$

$$intra\text{-}classe^{TF}_{minou,Felidae}\text{-}Norm = \frac{intra\text{-}classe^{TF}_{minou,Felidae}}{\arg\max intra\text{-}classe^{TF}} = \frac{0.33}{0.33} = 1$$

$$inter\text{-}classes^{ICF}_{minou,Felidae}\text{-}Norm = \frac{inter\text{-}classes^{ICF}_{minou,Felidae}}{\arg\max inter\text{-}classes^{ICF}} = \frac{1.58}{1.58} = 1$$

$$inter\text{-}classes^{IDF}_{minou,Felidae}\text{-}Norm = \frac{inter\text{-}classes^{IDF}_{minou,Felidae}}{\arg\max inter\text{-}classes^{IDF}} = \frac{2.81}{2.81} = 1$$

$$inter\text{-}classes^{ITF}_{minou,Felidae}\text{-}Norm = \frac{inter\text{-}classes^{ITF}_{minou,Felidae}}{\arg\max inter\text{-}classes^{ITF}} = \frac{5}{5.21} = 0.96$$

Comme nous pouvons le constater dans le tableau 4.9, les pondérations se positionnent maintenant les unes par rapport aux autres et non plus en fonction du nombre de classes, de documents ou de descripteurs.

Table 4.9: Pondérations normalisées

		intra-classe		*inter-classes*		
Classe	Descripteur	DF-Norm	TF-Norm	ICF-Norm	IDF-Norm	ITF-Norm
Bovidae	animal	1	0.69	0	0	0.46
	compagnie	0.33	0.23	0	0	0.46
	bœuf	0.67	0.46	1	1	1
	génisse	0.33	0.23	1	1	1
	taureau	0.33	0.23	1	1	1
	vachette	0.33	0.23	1	1	1
	yack	0.67	0.92	1	1	1
Canidae	animal	1	0.50	0	0	0.42
	chien	0.33	0.17	1	1	0.96
	chienne	0.33	0.17	1	1	0.96
	chiot	0.67	0.50	1	1	0.96
	compagnie	1	0.50	0	0.17	0.51
	medor	0.33	0.17	1	1	0.96
	milou	0.33	0.17	1	1	0.96
	pif	0.33	0.17	1	1	0.96
	rantanplan	0.33	0.17	1	1	0.96
	toutou	0.67	0.50	1	1	0.96
Felidae	animal	1	0.50	0	0	0.42
	chat	0.33	0.17	1	1	0.96
	compagnie	1	0.50	0	0.17	0.51
	matou	1	0.67	1	1	0.96
	minet	0.33	0.17	1	1	0.96
	minou	0.67	1	1	1	0.96

À partir de ces pondérations normalisées, nous présentons dans la section suivante deux mesures locales qui permettront de constituer le poids global tout en respectant l'équilibre entre les poids intra-classes et les poids inter-classes.

4.4.3 Mesure globale

La solution la plus naturelle serait de réaliser une moyenne des pondérations $intra\text{-}classe_{ij}^{DF}$ et $intra\text{-}classe_{ij}^{TF}$ au sein de la mesure intra-classe locale (que nous nommons $intra\text{-}classe^{loc}$) telle que défini dans la formule 1 et une moyenne des pondérations $inter\text{-}classes_{ij}^{ICF}$, $inter\text{-}classes_{ij}^{IDF}$ et $inter\text{-}classes_{ij}^{ITF}$ au sein de la mesure inter-classes locale (que nous nommons $inter\text{-}classes^{loc}$) telle que défini dans la formule 2

Formule 1 ($intra\text{-}classe^{loc}$)

$$intra\text{-}classe^{loc} = \frac{1}{2} \times intra\text{-}classe_{ij}^{DF} + \frac{1}{2} \times intra\text{-}classe_{ij}^{TF}$$

Formule 2 ($inter\text{-}classes^{loc}$)

$$inter\text{-}classes^{loc} = \frac{1}{3} \times inter\text{-}classes_{ij}^{ICF} + \frac{1}{3} \times inter\text{-}classes_{ij}^{IDF} + \frac{1}{3} \times inter\text{-}classes_{ij}^{ITF}$$

La pondération globale w_{ij}^{glob} (Définition n°7) est calculée à partir des cinq pondérations et est obtenue en multipliant la mesure intra-classe locale et la mesure inter-classes locale.

Définition 7 (w_{ij}^{glob})

w_{ij}^{glob} : Pondération globale du descripteur t_i de la classe C_j calculée à partir des cinq pondérations $intra\text{-}classe_{ij}^{DF}$, $intra\text{-}classe_{ij}^{TF}$, $inter\text{-}classes_{ij}^{ICF}$, $inter\text{-}classes_{ij}^{IDF}$ et $inter\text{-}classes_{ij}^{ITF}$.

Avec : $w_{ij}^{glob} = intra\text{-}classe^{loc} \times inter\text{-}classes^{loc}$

Pour illustrer, nous reprenons notre exemple et nous calculons le poids global $w_{minou,Felidae}^{glob}$.

Exemple 7 ($w^{glob}_{minou,Felidae}$)

$w^{glob}_{minou,Felidae} = (\frac{1}{2} \times 0.67 + \frac{1}{2} \times 1) \times (\frac{1}{3} \times 1 + \frac{1}{3} \times 1 + \frac{1}{3} \times 0.96) = (0.335 + 0.5) \times (0.33 + 0.33 + 0.32) = 0.84 \times 0.98 = 0.82$

4.4.4 Mesure globale paramétrique

Nous proposons d'affiner encore notre proposition afin de s'adapter au mieux aux différents types de corpus. En effet, le résultat obtenu pour une pondération donnée va être influencé par la composition du corpus, comme par exemple :

- le nombre de classes
- le nombre de documents par classe
- le nombre de descripteurs par document
- le déséquilibre entre la taille des documents, du nombre de documents ou de classes

Nous pensons que les proportions ($\frac{1}{2}$-$\frac{1}{2}$ pour les pondérations intra-classes, $\frac{1}{3}$-$\frac{1}{3}$-$\frac{1}{3}$ pour les pondérations inter-classes), plutôt que d'être fixées, peuvent être généralisées sous forme de combinaisons linéaires (Définition n°8). Ceci à l'avantage d'accentuer (resp de minimiser) le poids de chacun des composantes selon les caractéristiques des données. Les paramètres et leur influence seront étudiés dans le Chapitre 5.

Définition 8 ($intra\text{-}classe^{loc}$ et $inter\text{-}classes^{loc}$ paramétriques)

$$intra\text{-}classe^{loc} = \alpha \times intra\text{-}classe^{TF}_{ij} + (1 - \alpha) \times intra\text{-}classe^{DF}_{ij}$$

$$inter\text{-}classes^{loc} = \beta \times inter\text{-}classes^{ICF}_{ij} + \omega \times inter\text{-}classes^{IDF}_{ij} + (1 - \beta - \omega) \times inter\text{-}classes^{ITF}_{ij}$$

Ces 3 paramètres α, β et ω doivent cependant respecter un certain nombre de contraintes.

1. α, β et ω doivent être compris entre 0 et 1.
2. $\beta + \omega \leq 1$ afin que la somme des coefficients β, ω et $1 - \beta - \omega$ soit toujours égale à 1.

Le respect de ces contraintes permet de conserver l'équilibre entre les poids intra-classes et les poids inter-classes dans la pondération globale (le calcul du poids global étant toujours le produit des deux poids locaux).
Par exemple, nous calculons le poids global $w^{glob}_{(minou,Felidae)}$ avec les paramètres ($\alpha = 0.6$, $\beta = 0.6$ et $\omega = 0$), puis avec les paramètres ($\alpha = 0.5$, $\beta = 0.33$ et $\omega = 33$) et enfin avec les paramètres ($\alpha = 0$, $\beta = 0$ et $\omega = 0$).

Exemple 8 ($w^{glob}_{minou,Felidae}$)

$\alpha = 0.6$, $\beta = 0.6$ et $\omega = 0$
$w^{glob}_{minou,Felidae} = (0.6 \times 0.67 + (1 - 0.6) \times 1) \times (0.6 \times 1 + 0 \times 1 + (1 - 0.6 - 0) \times 0.96) = (0.6 + 0.27) \times (0.6 + 0 + 0.38) = 0.87 \times 0.98 = 0.85$

$\alpha = 0.5$, $\beta = 0.33$ et $\omega = 33$
$w^{glob}_{minou,Felidae} = (0.5 \times 0.67 + (1 - 0.5) \times 1) \times (0.33 \times 1 + 0.33 \times 1 + 0.33 \times 0.96) = (0.34 + 0.5) \times (0.33 + 0.33 + 0.32) = 0.84 \times 0.98 = 0.82$

$\alpha = 0$, $\beta = 0$ et $\omega = 0$
$w^{glob}_{minou,Felidae} = (0 \times 0.67 + (1 - 0) \times 1) \times (0 \times 1 + 0 \times 1 + 1 \times 0.96) = (0 + 1) \times (0 + 0 + 0.96) = 1 \times 0.96 = 0.96$

Pour compléter notre étude de cas, nous reprenons notre corpus exemple et nous calculons les poids globaux de l'ensemble des descripteurs du corpus en utilisant les paramètres $\alpha = 0.6$, $\beta = 0.6$ et $\omega = 0$ dans le tableau 4.10 pour l'ensemble des descripteurs du corpus.

Table 4.10: Pondérations normalisées et paramétrées

Classe	Descripteur	$intra\text{-}classe^{loc}$	$inter\text{-}classes^{loc}$	w^{glob}
	paramètres	$\alpha = 0.6$	$\beta = 0.6, \omega = 0$	
Bovidae	animal	0.82	0.18	0.15
	compagnie	0.27	0.18	0.05
	bœuf	0.54	1	0.54
	génisse	0.27	1	0.27
	taureau	0.27	1	0.27
	vachette	0.27	1	0.27
	yack	0.82	1	0.82
Canidae	animal	0.70	0.17	0.12
	chien	0.23	0.98	0.23
	chienne	0.23	0.98	0.23
	chiot	0.57	0.98	0.56
	compagnie	0.70	0.21	0.14
	medor	0.23	0.98	0.23
	milou	0.23	0.98	0.23
	pif	0.23	0.98	0.23
	rantanplan	0.23	0.98	0.23
	toutou	0.57	0.98	0.56
Felidae	animal	0.70	0.17	0.12
	chat	0.23	0.98	0.23
	compagnie	0.70	0.21	0.14
	matou	0.80	0.98	0.79
	minet	0.23	0.98	0.23
	minou	0.87	0.98	0.85

Avec cette première contribution (Définition 8), nous pensons qu'en définissant des pondérations adaptées (i.e. les valeurs α, β et ω), nous pouvons améliorer les résultats obtenus pour traiter les problèmes de classification. Les expérimentations menées dans le Chapitre 5 confirment l'intérêt de notre approche globale et paramétrique.

4.5 Bilan et discussions

Dans ce chapitre, nous venons de définir un processus de pondération reposant sur 3 étapes, le calcul des pondérations (étape 1), la normalisation (étape 2) et enfin le calcul du poids global (étape 3). Les étapes 1 et 2 sont rapides à calculer et l'étape 3 rend les pondérations résilientes aux caractéristiques du corpus, ce qui permet d'être utilisé sur tout type de corpus. Notre approche est intuitive et elle permet d'obtenir un classement des descripteurs en fonction de leur importance au sein de la classe, mais aussi au regard du reste du corpus. Dans notre exemple les descripteurs "yack" et "bœuf" sont plus représentatifs de la classe "Bovidae" que les descripteurs "animal" et "compagnie". Enfin, notre proposition permet de comparer l'importance d'un même descripteur au sein de différentes classes du corpus. Dans notre exemple, le descripteur "compagnie" est plus représentatif des classes "Felidae" et "Canidae" que de la classe "Bovidae".

Nous proposons d'utiliser et d'intégrer les pondérations au sein d'une approche de classification supervisée de documents. Fixer les valeurs des paramètres α, β et ω les plus appropriées au corpus est difficile en raison du nombre de possibilités. Nous discutons des solutions et nous proposons une nouvelle approche dans la seconde partie de ce manuscrit pour détecter les meilleurs paramètres.

Pour le moment, nous conviendrons que les différents paramètres seront testés empiriquement. Même si cette approche peut paraitre consommatrice en temps, elle garantit des résultats optimaux et nous permet d'étudier leur impact sur les résultats. Nous présentons les expérimentations et les résultats obtenus dans le chapitre suivant.

Chapitre 5

Expérimentations de la mesure w_{ij}^{glob}

Dans ce chapitre, nous présentons les expérimentations réalisées afin d'étudier les performances de notre proposition et plus particulièrement de la mesure globale.

L'objectif de ces travaux était initialement l'amélioration des algorithmes de classification pour un corpus précis soumis par la société Itesoft. Nous présentons les spécificités de ce corpus (Section 5.1). Nous discutons de l'intégration de notre approche dans des algorithmes de classifications supervisées (Section 5.2) ainsi que les méthodes et algorithmes de la littérature ayant servis à évaluer la pertinence de notre approche (Section 5.3). Puis nous présentons les résultats obtenus pour le corpus Itesoft (5.4). Afin de compléter nos analyses, nous introduisons des corpus supplémentaires et présentons des séries d'expérimentations visant à mieux analyser le comportement de notre proposition en fonction de la composition des corpus (Section 5.5). Enfin, nous concluons ce chapitre et cette première partie avec une discussion globale sur notre première contribution (Chapitre 6).

5.1 Corpus Itesoft

Ces travaux ont été initialement développés pour traiter un type de corpus caractéristiques de la société ITESOFT. Ce corpus est composé de courriers "océrisés" qui doivent être classés dans des catégories prédéfinies. Dans la suite de ce manuscrit, nous y ferons référence comme étant le corpus "*Itesoft*".
Pour des raisons de confidentialité, nous ne détaillerons pas le contenu des lettres et nous nous contenterons de reproduire ici un exemple anonymisé par nos soins.

Exemple 1 (Exemple de courrier Itesoft)

Mr et Mme XXXXXX N° Client S XXXXXX 69140 RILLIEUX LA PAPE XXXXXX le 14 novembre 2008 Lettre recommandée avec accusé de réception N. abonné XXXXXX Service abonnement XXXXXX Madame, Monsieur, Je vous prie de bien vouloir prendre note, par la présente. de ma demande de résiliation de l' abormement référencé ci-dessus J' ai bien pris note que cette .résiliation prendra effet à la date anniversaire soit au 1er février 2009. Vous en souhaitant bonne réception et dans l' attente de vous lire afin de coimaître les modalités de cette résiliation. je vous remercie d ? avance. Je vous prie d' agréer. Madame, Monsieur, mes sincères salutations. Mme XXXXXX

Ces documents présentent les caractéristiques suivantes :

- Ils contiennent des fautes, introduites soit par le système d'océrisation, soit par le rédacteur du message.
- La ponctuation est mal retranscrite (ajout ou suppression de ponctuation) rendant les outils de Traitement Automatique des Langues mal adaptés.
- Le corps du message est mélangé aux en-têtes, formules de politesse et parfois même aux pièce-jointes. Le nombre de descripteurs utiles (i.e. porteur d'informations) ne représente pas une large majorité des descripteurs du document. Nos pondérations ayant été pensées pour être résistantes à ce type de descripteurs, nous avons décidé de conserver comme faisant partie du message.

Afin de limiter l'espace de recherche, nous avons choisi de supprimer les mots outils (stopwords) et les descripteurs inférieurs à 3 caractères et, pour des raisons de performance et de fiabilité dans l'interprétation des résultats, nous n'avons conservé que les classes composées à minima de 45 documents.
Le Tableau 5.1 présente les caractéristiques du corpus *Itesoft*.

Table 5.1: Volumétrie des différents corpus

Corpus	Nb de classes	Nb de documents	Nb de descripteurs	Nb de descripteurs distincts
Itesoft	6	1 273	124 538	23 824

À partir de ce corpus, nous souhaitons construire un modèle nous permettant de prédire l'affectation d'une lettre à une catégorie, par exemple de déterminer si le courrier entrant est une demande de résiliation ou une simple question sur un abonnement en cours. Pour ce faire, nous avons intégré nos pondérations dans des approches de classifications supervisées et nous discutons de cette intégration dans la section suivante.

5.2 Intégration des mesures dans un contexte d'apprentissage supervisé

Comme nous l'avons présenté dans le chapitre introductif de cette partie, il existe un grand nombre de classifieurs qui reposent sur des approches différentes. Certains se situent au niveau des documents (par exemple, la méthode des plus proches voisins ou celle des séparateurs à vaste marge) pour construire leur modèle là où d'autres se basent sur des informations agrégées au niveau des classes (Classifieurs bayésiens, Méthodes du centroïde).

Nos nouvelles méthodes de pondération, en définissant le poids d'un descripteur pour une classe donnée, peuvent être utilisées dans les approches du second groupe qui sont fondées sur des descripteurs pondérés au niveau des classes ce qui rend nos mesures bien adaptées à ce type d'algorithme. De plus, elles présentent les avantages suivants :

1. leur facilité de mise en œuvre les rend bien adaptées pour la classification automatique de documents ;
2. les modèles obtenus sont faciles à interpréter et à valider pour un utilisateur final.

Dans les sections suivantes, nous présentons rapidement deux approches retenues avec, pour commencer, les méthodes basées sur les centroïdes.

5.2.1 Méthodes basées sur les centroïdes

Les méthodes basées sur les centroïdes sont des variantes des modèles linéaires [61]. La méthode traditionnelle peut être vue comme une spécialisation de la méthode Rocchio [131] utilisée dans de nombreux travaux de classification textuelle, par exemple dans [58, 61]. Dans ce type d'approche, chaque classe est considérée selon le modèle vectoriel de Salton [138]. Chacune des classes est représentée par un vecteur de descripteurs. $\overrightarrow{C_j} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ est la représentation de la classe j où w_{ij} est le poids du descripteur 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 descripteurs ($\overrightarrow{d} = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$) 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. Une définition de l'approche basée sur les centroïdes appliquée à la classification textuelle est donnée dans [89].

Généralement il y a deux méthodes pour créer le prototype de la classe C_j, soit utiliser la moyenne arithmétique des documents de la classe C_j (méthode AAC : *Arithmetical Average Centroid*), soit utiliser la somme des poids des descripteurs contenus dans les documents de la classe C_j (méthode CGC : *Cumuli Geometric Centroid*).

Ce type de classifieur est facile à implémenter et donne de bons résultats [57, 29, 133]. Par exemple, des travaux précédents ont montré que les méthodes basées sur les centroïdes donnaient des résultats satisfaisants en terme d'exactitude, mais aussi en temps d'exécution [50, 167]. Cependant dans [163], les auteurs montrent que les approches basées sur les centroïdes sont significativement moins performantes que d'autres approches (par exemple SVM). Une des raisons avancées repose sur de "mauvaises" valeurs initiales du centroïde. Plusieurs méthodes ont été proposées pour améliorer itérativement la justesse du centroïde (Dragpushing method [164], Hypothesis margin method [165], Weight Ajustement Method [146]).
Les résultats de ces approches adaptatives sont meilleurs et comparables aux résultats obtenus par SVM [164, 165].

Les résultats dépendent fortement de la pondération des descripteurs. La plupart des travaux se basent sur une pondération $tf.idf$ [2] mais d'autres pondérations sont utilisées dans [36, 167, 33]. Enfin, plusieurs méthodes pour réduire la dimension des centroïdes ont été utilisées ; par exemple la suppression des stopwords ou l'utilisation des racines [186], le clustering de descripteurs [96] ou encore la fréquence de documents [187].

Parmi les méthodes basées sur les centroïdes, l'approche *Class-Feature-Centroid* est un modèle récent présenté dans [49]. Le poids d'un descripteur est défini par le produit entre la fréquence de document (poids intra-classe) et la fréquence inverse du nombre de classes contenant le descripteur (poids inter-classes) selon la formule 3.

Formule 3 (Pondération utilisée par Guan [49])

$$w_{i,j}^{CFC} = b^{\frac{DF_{ti}^{j}}{|d_j|}} \times \log(\frac{|C|}{|C_{ti}|})$$

Où :
- b est un paramètre supérieur ou égal à 1,
- DF_{ti}^{j} correspond au nombre de documents de la classe C_j qui contiennent le descripteur t_i,
- $|d_j|$ correspond au nombre total de documents de la classe C_j,
- $|C|$ correspond au nombre total de classes du corpus,
- $|C_{ti}|$ correspond au nombre de classes du corpus qui contient le descripteur t_i

Nous pouvons faire un parallèle entre notre approche et cette formule en utilisant les pondérations $intra\text{-}classe^{DF}$ et $inter\text{-}classes^{ICF}$ dans leur forme non normalisée. Avec les notations introduites dans ce manuscrit, la formule utilisée

par Guan pourrait s'écrire

$$w_{i,j}^{CFC} = b^{intra\text{-}classe^{DF}} \times inter\text{-}classes^{ICF}$$

Les expérimentations menées dans [49] montrent que l'approche *Class-Feature-Centroid* est plus efficace et robuste avec les données clairsemées (Sparse Data). En outre, les classifieurs basés sur les *Class-Feature-Centroid* sont rapides, efficaces et faciles à déployer [111] mais présentent les inconvénients suivants :

- ils sont peu efficaces pour traiter des problèmes de classification binaire (et globalement avec un nombre restreint de classes), car les descripteurs trop spécifiques (comme les noms de personnes) peuvent devenir discriminants. Une trop grande importance est apportée au poids inter-classes.
- ils sont peu efficaces pour traiter avec les classes déséquilibrées (le biais tendant vers la classe majoritaire). L'approche *Class-Feature-Centroid* utilise le cosinus dénormalisé. Les descripteurs discriminants (ceux qui appartiennent à une seule classe) vont influencer grandement la décision et donc les classes les plus grandes seront avantagées.

Les auteurs utilisent le cosinus dénormalisé pour le test, mais d'autres mesures de similarité ont été utilisées dans la littérature, comme par exemple, le coefficient de corrélation de Pearson [127] ou encore la distance euclidienne [22].

Afin d'évaluer notre contribution, nous décidons d'intégrer nos pondérations comme poids du centroïde $C_j = \{w_{1j}^{glob}, w_{2j}^{glob}, ..., w_{|L|j}^{glob}\}$ où w_{ij}^{glob} est le poids global calculé pour le i-ème descripteur de la j-ème classe.

Nous décidons aussi d'intégrer le poids global w_{ij}^{glob} dans une approche de type Naive Bayes.

5.2.2 Classifieurs bayésiens

Le classifieur Naive Bayes est un classifieur de type probabiliste défini dans [90] très utilisé pour la classification de textes, 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 descripteurs t_i appartienne à la classe C_j est donnée par $P(d \in C_j) = P(C_j) \prod_i P(t_i|C_j)$. Le modèle Naive Bayes Multinomial est réputé pour offrir de bonnes performances pour traiter les problèmes de classification textuelle [141]. La principale difficulté réside dans la modélisation des hypothèses. Dans la pratique, la plupart des approches ont recours à des modifications pour corriger ces hypothèses comme par exemple la modification apportée pour éviter qu'un descripteur apparaisse 0 fois. Des combinaisons de modifications ont été présentées comme des solutions robustes pour la classification de texte [126] par exemple dans [73] où les auteurs utilisent une méthode de recherche aléatoire pour optimiser les modifications ap-

portées au modèle Naive Bayes Multinomial.

Naive Bayes est rapide et facile à implémenter, mais offre néanmoins des performances en général moindre. Dans [126] les auteurs étudient les raisons de ces moins bonnes performances et proposent le modèle *Transformed Weight Normalized Complement Naive Bayes* qui se base sur le modèle Naive Bayes Multinomiale et est très proche de celui-ci, la principale différence revient dans l'utilisation des autres classes du corpus en complément. Dans *Transformed Weight Normalized Complement Naive Bayes*, la probabilité est remplacée par un poids $\theta_{ci} = \frac{1+\sum_{j=1,j\neq c}^{|C|} d_{ij}}{N+\sum_{j=1,j\neq c}^{|C|}\sum_{k=1}^{N} d_{kj}}$.
Transformed Weight Normalized Complement Naive Bayes a montré des résultats proches de SVM. Néanmoins, les adaptations avancées par le modèle sont remises en cause dans [72] pour qui toutes les transformations ne sont pas nécessaires.

Nous proposons d'intégrer nos pondérations dans une approche similaire. Après avoir calculé $C_j = \{w_{1j}, w_{2j}, ..., w_{|L|j}\}$ où $w_{i,j}$ est le poids du $i^{ème}$ descripteur de la classe C_j, nous estimons un score pour un document non étiqueté d et une classe C_j selon $S(d \in C_j) = \prod_i (w_{i,j} + 1)^n$ où n correspond au nombre d'occurrences du descripteur dans le document à classifier. Ajouter 1 permet de prévenir le cas où le descripteur n'apparait pas dans la classe, afin d'éviter que le score soit égal à zéro. La classe C_j qui sera attribuée au document d sera celle pour laquelle le score $S(d \in C_j)$ sera le plus élevé.

Afin d'évaluer la pertinence de nos propositions, nous les comparons avec certains algorithmes de la littérature et nous les présentons ainsi que la méthode même de comparaison dans la section suivante.

5.3 Protocole expérimental

5.3.1 Algorithmes de comparaison

Il est généralement acquis qu'il n'est pas possible d'atteindre un taux de 100% de classifications correctes sur un corpus réel. L'objectif consiste alors à maximiser ce pourcentage. De plus, comme nous l'avons dit précédemment, il n'existe pas dans la littérature de méthodes de classifications qui fassent consensus pour traiter les problèmes indépendamment du corpus et les comparer toutes est impossible vu le nombre de méthodes développées. Nous avons choisi de tester un ensemble d'algorithmes de la suite logicielle Weka [1]. En choisissant un large spectre d'algorithmes couvrant les différentes familles de méthodes, nous estimons pouvoir obtenir une baseline fiable et facilement interprétable de comparaison.
Les différents algorithmes implémentés dans Weka qui ont servi de baseline sont :

1. http ://www.cs.waikato.ac.nz/ml/weka/

- Deux implémentations de SVM
  - *SMO*, qui utilise un noyau polynomial [117]
  - *LibSVM*, qui utilise un noyau linéaire [25]
- Deux arbres de décision
  - *LadTree* [54]
  - *J48* [122]
- Quatre implémentations de classifieurs bayésiens
  - *NaiveBayes* [63]
  - *NaiveBayesMultinomial* [98]
  - *ComplementNaiveBayes* [126] qui implémente le modèle *Transformed Weight Normalized Complement Naive Bayes*
  - *Discriminative Multinominal Naive Bayes - DMNB* [160]

D'autres modèles ont été testés (RandomTree, RepTree, ZeroR, OneR, IB1 et deux LibSVM, l'un avec un noyau polynomial et l'autre avec un noyau RBF - Radial Basis Function), mais les résultats ne seront pas présentés ici en raison des faibles scores obtenus. Une pondération classique $TF\text{-}IDF$ est utilisée pour pondérer les documents en amont de Weka lorsque nécessaire.

Avant de passer à la présentation des résultats, il convient tout d'abord de discuter de la notion d'évaluation.

5.3.2 Critères d'évaluation

Nous avons précisé que la pondération la plus pertinente était celle permettant d'obtenir les meilleures performances en classification. Il convient alors de définir ce qu'est une bonne classification. Est-ce une classification rapide ? Est-ce une classification sans erreur ? Ou encore une classification exhaustive de l'ensemble des problèmes à classer ? Nous verrons au travers des expérimentations que le modèle le plus performant peut varier selon le critère observé pour un même problème de classification.

Dans ce manuscrit, nous décidons d'évaluer la qualité des modèles au moyen de la F-mesure qui est utilisée dans la plupart des travaux. La F-mesure repose sur deux métriques, la Précision et le Rappel. Le Rappel est défini en divisant le nombre de documents bien classés par le nombre de documents à classer, un Rappel de 100% indique que tous les documents ont été correctement classés. La Précision est définie en divisant le nombre de documents bien classés par rapport au nombre de documents classés, une Précision de 100% indique que le système est fiable dans la réponse qu'il donne. Un système parfait obtiendrait 100% de Précision et 100% de Rappel. Dans les faits, un compromis est généralement réalisé entre une bonne Précision et un bon Rappel. Et pour cela la F-mesure est utilisée. Elle correspond à la moyenne harmonique de la Précision et du Rappel ($\frac{2 \times Rappel \times Précision}{Rappel + Précision}$).

Par ailleurs, dans ce manuscrit, nous calculons la F-mesure selon la vision Micro et la vision Macro [108]. Dans la première, le Rappel (resp la Précision) est calculé pour l'ensemble des classes. En revanche, pour la vision Macro, le Rappel (resp la Précision) est d'abord calculé indépendamment pour chaque classe, puis la moyenne des Rappels (resp des Précisions) est effectuée. La vision Macro, en effectuant la moyenne, va donner une même importance aux classes dans la mesure finale alors que la vision Micro va donner un poids plus important aux classes majoritaires. Utiliser les deux en complément permet d'affiner la lecture des résultats. Prenons un exemple (tableau 5.2) pour illustrer les différents calculs.

Table 5.2: Calcul de la Précision et du Rappel

Classe	Nb doc à classer	Nb doc attribués à la classe	Nb doc correctement attribués à la classe
Bovidae	10	10	9
Canidae	50	10	1
Felidae	100	140	90
Total	160	160	100

Par exemple pour le calcul de la Précision, dans la vision Micro, nous calculons le nombre total de documents correctement attribués à une classe par rapport au nombre total de documents attribués à une classe ($\frac{100}{160} = 0.625$).
Dans la vision Macro, nous calculons d'abord la Précision de chaque classe indépendamment (Précision Bovidae : $\frac{9}{10} = 0.9$, Précision Canidae : $\frac{1}{10} = 0.1$, Précision Canidae : $\frac{90}{140} = 0.64$) avant d'en faire la moyenne (Micro Précision : $\frac{0.9+0.1+0.64}{3} = 0.55$).
De même pour le Rappel, dans la vision Macro, nous calculons le nombre total de documents correctement attribués à une classe par rapport au nombre total de documents à attribuer ($\frac{100}{160} = 0.625$).
Dans la vision Macro, nous allons d'abord calculer le Rappel de chaque classe indépendamment (Rappel Bovidae : $\frac{9}{10} = 1$, Rappel Canidae : $\frac{1}{50} = 0.02$, Rappel Canidae : $\frac{90}{100} = 0.9$) avant d'en faire la moyenne (Macro Rappel : $\frac{0.9+0.02+0.9}{3} = 0.61$).

La Micro F-mesure est obtenue en faisant la moyenne harmonique de la Micro Précision et du Micro Rappel et la Macro F-mesure est obtenue en faisant la moyenne harmonique de la Macro Précision et du Macro Rappel.

Bien que la rapidité d'exécution soit aussi un critère recherché, celui-ci est trop dépendant de la machine sur laquelle est exécuté le traitement et de la qualité des optimisations développées dans les prototypes pour être un indicateur fiable. De plus, les expérimentations réalisées ont eu des temps de traitement assez similaires et raisonnables pour que ce critère ne soit pas jugé comme crucial dans la suite de ce manuscrit. Chacun des algorithmes a été testé en validation croisée (*3-fold cross validation*). Le nombre de 3 itérations a été retenu afin d'avoir un nombre suffisant de données en apprentissage et en test. Les résultats présentés ci-après correspondent à la moyenne des 3 itérations.

5.3.3 Paramètres testés

Notre proposition repose sur un ensemble de paramètres. Comme nous l'avons présenté lors de l'introduction des paramètres (Section 4.4.3), les 3 paramètres α, β et ω doivent respecter un certain nombre de contraintes (α, β et ω doivent être compris entre 0 et 1, $\beta + \omega \leq 1$). Ainsi, nous obtenons autant de résultats que de variations de paramètres pour notre proposition intégrée dans Naive Bayes et encore autant pour nos pondérations intégrées dans l'approche Class-Feature-Centroid, en plus des 8 résultats obtenus avec les 8 algorithmes de comparaison retenus (Section 5.3.1).

A noter que α étant indépendant du couple β et ω, si nous choisissons 2 valeurs pour α, par exemple [0, 1], et 3 valeurs pour le couple (β, ω), par exemple [(1,0), (0,1), (0,0)], nous obtiendrons 6 cas de figure différents (2 × 3) :

1. $\alpha = 1$, β=1, $\omega = 0$
2. $\alpha = 1$, β=0, $\omega = 1$
3. $\alpha = 1$, β=0, $\omega = 0$
4. $\alpha = 0$, β=1, $\omega = 0$
5. $\alpha = 0$, β=0, $\omega = 1$
6. $\alpha = 0$, β=0, $\omega = 0$

Lors de nos expérimentations, le défi consistait à trouver un compromis entre restreindre le nombre de valeurs de paramètres tout en couvrant l'ensemble des possibles. C'est pourquoi, nous avons décidé de retenir 5 valeurs pour α [0, 0.25, 0.5, 0.75, 1] et 12 couples de valeurs pour les paramètres β et ω [(1,0), (0,1), (0,0), (0.25,0.75), (0.25,0), (0.75,0.25), (0.75,0), (0,0.75), (0,0.25), (0.5,0.5), (0.5,0), (0,0.5), (0.33,0.33)].

Ainsi pour le corpus *Itesoft*, nous obtenons un total de 128 évaluations :
- 8 évaluations pour les algorithmes de comparaison
- 60 évaluations (5 × 12) pour notre proposition intégrée dans l'approche Naive Bayes
- 60 évaluations pour notre proposition intégrée dans l'approche Class-Feature-Centroid.

Nous présentons les résultats dans la section suivante.

5.4 Résultats

Ces travaux ont été développés dans l'objectif industriel de pouvoir traiter le corpus *Itesoft* réputé comme difficile. Nous présentons les résultats des expérimentations passées sur ce corpus dans le graphique 5.1 qu'il convient de détailler. Tout d'abord nous affichons chacune des mesures effectuées en fonction de la

Micro F-mesure (Axe des abscisses) et de la Macro F-Mesure (Axe des ordonnées). Les algorithmes de comparaison sont symbolisés par des losanges bleus, nos expérimentations menées avec une approche Naive Bayes sont symbolisées par un rond vert et celles menées avec une approche CFC par un tiret rouge. Deux cercles orange ont été apposés afin d'illustrer nos propos. Le premier, qui entoure un rond vert, indique les résultats Micro et Macro obtenus avec l'approche Naive Bayes et les paramètres $\alpha = 1$, $\beta = 0$ et $\omega = 0.75$. Le second, qui entoure un losange bleu, correspond aux Micro et Macro F-mesures obtenues avec l'algorithme NaiveBayes de Weka. Pour finir nous avons ajouté deux axes rouges qui se croisent aux maximums des Micro et Macro F-mesures obtenues avec les algorithmes de comparaison. Elles permettent de diviser le graphe en 4 parties (numérotées de 1 à 4). Les mesures présentes dans les cases 1 et 2 ont une Macro F-mesure supérieure à la meilleure Macro F-mesure relevée pour les algorithmes de comparaison alors que celles présentent dans les cases 1 et 3 ont une Micro F-mesure supérieure à la meilleure Micro F-mesure relevée pour les algorithmes de comparaison. Dans la case 1, nous retrouvons les mesures ayant obtenues à la fois une meilleure Micro F-mesure et une meilleure Macro F-mesure que les maximums constatés avec les algorithmes de comparaison. C'est le cas le plus intéressant puisque nous retrouvons dans cette case nos pondérations avec les paramètres nous permettant d'obtenir de meilleurs résultats à la fois en Micro et à la fois en Macro que les algorithmes de la littérature utilisés. À l'inverse nous retrouvons dans la case 4, les mesures à la fois moins performantes en Micro et Macro F-mesure que les maximums constatés avec les algorithmes de comparaison.

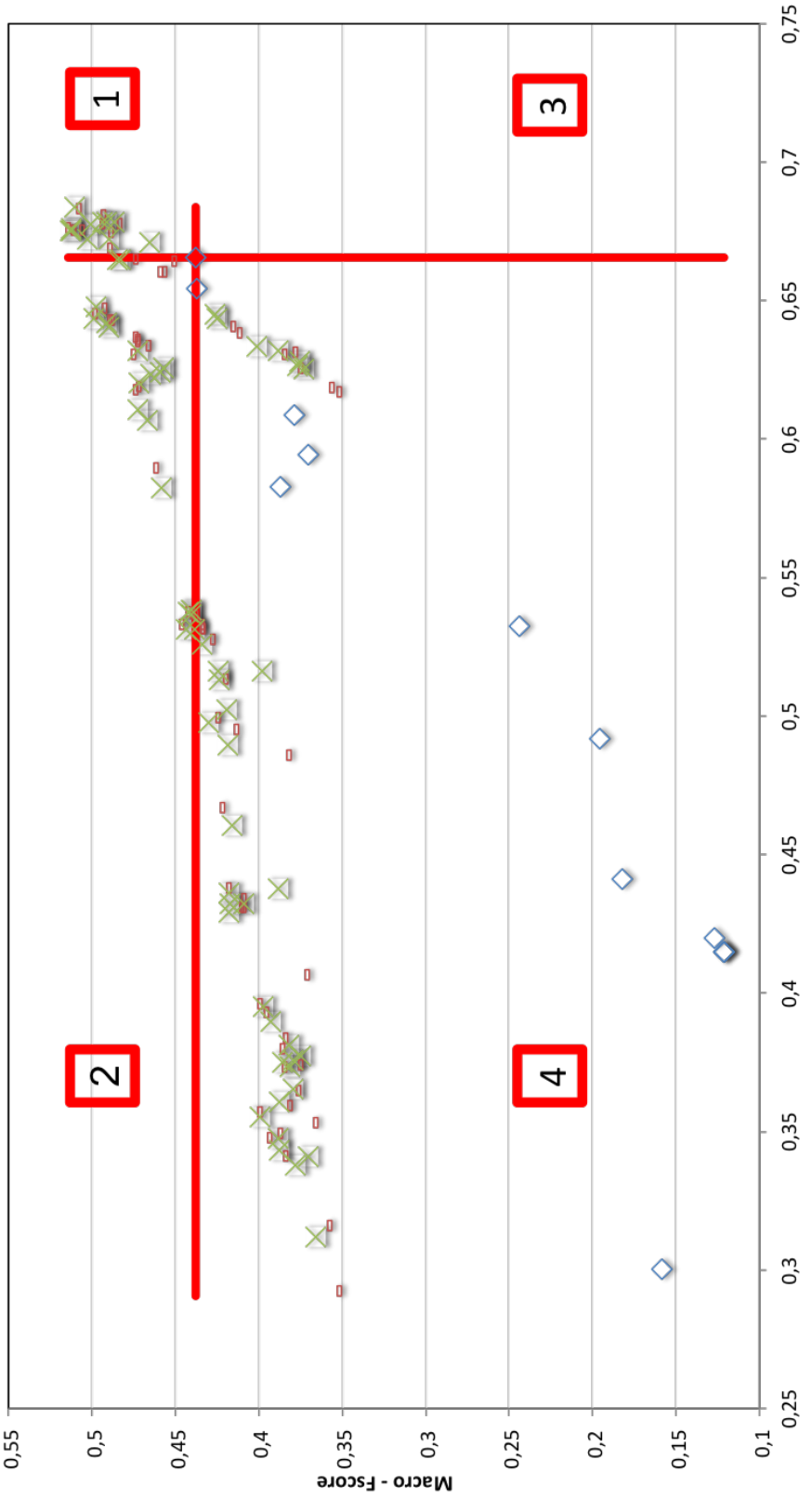


Figure 5.1: Resultats Corpus Itesoft

Avant de revenir plus en détails sur les résultats, nous pouvons déjà faire ressortir quelques confirmations, mais aussi quelques points intéressants de nos propositions :

1. Il existe un ensemble de paramètres pour nos modèles qui permettent d'obtenir de meilleurs résultats par rapport aux autres algorithmes. Ces résultats sont meilleurs à la fois en Micro F-mesure mais aussi en Macro F-mesure. Notre approche, avec les paramètres adaptés au corpus *Itesoft*, permet de traiter le corpus de façon plus efficace à la fois dans sa globalité (Micro F-mesure) mais aussi pour chacune des classes prise individuellement (Macro F-mesure).
2. Les choix des paramètres ont un impact très important sur la qualité du modèle obtenu.
3. Le choix des algorithmes de comparaison a aussi un réel impact sur la qualité de la classification, conformément à ce qui était attendu.
4. Il n'y a pas de différence significative entre une approche Naive Bayes ou une approche CFC.
5. Nos propositions offrent de bien meilleurs résultats en vision Macro (Partie 1 et 2) ce qui prouve que notre modèle ne privilégie pas les classes les plus représentées au détriment des autres.

Nous présentons dans le tableau 5.3, un aperçu des résultats obtenus pour les 8 algorithmes de comparaison les plus performants ainsi que pour les 10 meilleurs résultats obtenus (en considérant la moyenne des deux F-mesures) avec l'approche Naive Bayes puis avec l'approche CFC. Un fond de couleur est utilisé pour différencier les différentes familles (bleu pour les algorithmes de comparaison, neutre pour les approches CFC et vert pour les approches Naive Bayes).

Table 5.3: Résultats détaillés obtenus sur le corpus *Itesoft*

Algorithme	α	β	ω	MicroF-mesure	MacroF-mesure	Moyenne Micro et Macro
Naive Bayes	0	0.75	0.25	0.684	0.511	0.597
CFC	0	0.75	0.25	0.681	0.508	0.595
CFC	0.25	0.333333	0.333333	0.674	0.514	0.594
Naive Bayes	0.25	0	0.5	0.676	0.512	0.594
Naive Bayes	0.25	0	0.75	0.675	0.513	0.594
Naive Bayes	0.25	0.333333	0.333333	0.675	0.513	0.594
CFC	0.25	0	0.75	0.674	0.511	0.592
CFC	0.25	0	0.5	0.674	0.506	0.590
Naive Bayes	0.25	0.75	0	0.678	0.501	0.589
Naive Bayes	0.25	0	0.25	0.672	0.502	0.587
CFC	0	0.25	0.75	0.679	0.493	0.586
Naive Bayes	0	0	1	0.678	0.494	0.586
CFC	0	0	1	0.676	0.494	0.585
Naive Bayes	0	0.25	0.75	0.678	0.491	0.585
Naive Bayes	0	0.5	0.5	0.678	0.487	0.582
Naive Bayes	0.25	0.5	0	0.672	0.490	0.581
CFC	0.25	0.75	0	0.673	0.489	0.581
CFC	0	0.5	0.5	0.676	0.484	0.580
CFC	0.25	0	0.25	0.667	0.489	0.578
CFC	0.25	0.75	0.25	0.644	0.498	0.571
SMO				0.666	0.438	0.552
DMNB				0.655	0.437	0.546
LadTree				0.609	0.379	0.494
NaiveBayes				0.583	0.387	0.485
j48				0.594	0.371	0.482
ComplementNaiveBayes				0.533	0.244	0.388
LibSVM				0.441	0.182	0.312
NaiveBayesMultinomial				0.415	0.121	0.268

Les meilleurs résultats sur le corpus *Itesoft* sont obtenus avec l'algorithme Naive Bayes et les paramètres ($\alpha = 0$, $\beta = 0.75$ et $\omega = 0.25$). L'écart avec le meilleur algorithme de comparaison est de 0.02 pour la Micro F-Mesure (0.684 vs 0.666) et 0.07 sur la Macro F-Mesure (0.511 vs 0.438). Les meilleurs paramètres pour l'approche CFC sont les mêmes et les résultats sont comparables. En plus des "meilleurs" paramètres, nous nous intéressons aux moins performants et nous présentons aussi dans le tableau 5.4, les 10 combinaisons de paramètres les moins performantes.

Table 5.4: Les 10 plus mauvais résultats obtenus sur le corpus *Itesoft*

Algorithme	α	β	ω	MicroF-mesure	MacroF-mesure	Moyenne Micro et Macro
SMO				0.666	0.438	0.552
DMNB				0.655	0.437	0.546
LadTree				0.609	0.379	0.494
NaiveBayes				0.583	0.387	0.485
j48				0.594	0.371	0.482
ComplementNaiveBayes				0.533	0.244	0.388
Naive Bayes	1	0.5	0.5	0.347	0.389	0.368
CFC	1	0.5	0.5	0.347	0.387	0.367
Naive Bayes	1	0.25	0.75	0.343	0.388	0.366
CFC	1	0	1	0.339	0.384	0.362
CFC	0.5	1	0	0.351	0.366	0.359
Naive Bayes	1	0	1	0.338	0.378	0.358
Naive Bayes	0.75	1	0	0.341	0.371	0.356
Naive Bayes	1	1	0	0.312	0.366	0.339
CFC	0.75	1	0	0.314	0.358	0.336
CFC	1	1	0	0.291	0.352	0.321
LibSVM				0.441	0.182	0.312
NaiveBayesMultinomial				0.415	0.121	0.268

D'après les tableaux 5.3 et 5.4, il semblerait que pour le corpus Itesoft, le choix du paramètre α soit plus prépondérant que le choix des paramètres β et ω et qu'une valeur faible permette d'obtenir de meilleurs résultats. Concrètement, cela indique que le poids *intra-classe*^{TF} est plus pertinent que le poids *intra-classe*^{DF} mais qu'il existe moins de certitude concernant les poids inter-classes. Afin de confirmer ces résultats, nous avons expérimenté notre proposition sur deux autres corpus que nous présentons dans la section suivante.

5.5 Corpus expérimentaux supplémentaires

En choisissant plusieurs types de corpus, notre objectif est de pouvoir mieux étudier le comportement de notre mesure et étudier la généricité de notre proposition pour d'autres types de corpus. Nous avons ainsi utilisé des corpus de langues variées (anglais et français) et issus d'environnements différents (dépêches de presses, tweets politiques). Nous avons également souhaité avoir des corpus assez différents (nombre de classes, de documents ou de descripteurs) de même que le déséquilibre du nombre d'éléments entre les classes elles-mêmes.

Le premier corpus, *Reuters-21578*[2], est fréquemment utilisé par la communauté pour évaluer la qualité des modèles. Il est composé d'un ensemble de dépêches écrites en anglais et mises à disposition par l'agence Reuters. Les dépêches sont regroupées dans différentes catégories par exemple "sucre", "huile" ou "or", etc. Les documents ont été étiquetés manuellement. Dans la suite de ce manuscrit, nous y ferons référence comme étant le corpus "*Reuters*".

2. http ://trec.nist.gov/data/reuters/reuters.html/

Le second corpus est un corpus que nous avons constitué lors du projet Polop [3] (Political Opinion Mining). Le projet Polop, que nous avons mené conjointement avec une équipe de recherche canadienne (Université d'Ottawa) a pour objectif d'analyser le comportement de communautés au cours du temps via le réseau social Twitter. Dans le cadre de ce projet, il s'agit plus particulièrement d'étudier comment le comportement de différentes communautés évolue au cours du temps. L'un des exemples d'applications illustrant tout particulièrement cette problématique est l'analyse de tweets de partis politiques notamment lors des élections présidentielles et législatives de 2012 en France. Le corpus constitué à cette occasion est composé de plus de 2 millions 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. Avec ce corpus, notre objectif est de déterminer pour un utilisateur le rattachement à tel ou tel parti politique en fonction de ses tweets. Dans la suite de ce manuscrit, nous y ferons référence comme étant le corpus "*Polop*".

Pour chaque corpus, nous avons appliqué les mêmes prétraitements que sur le corpus *Itesoft* (suppression des stopwords, des descripteurs inférieurs à 3 caractères et conservation des classes composées à minima de 45 documents). De plus, certains documents du corpus *Reuters* sont multiclasses. Afin de conserver un processus qui soit indépendant du corpus lors des expérimentations et pour ne pas introduire de biais dans les évaluations, ces documents ont été retirés. Cela représente :

- 39 catégories pour le corpus *Reuters*.
- 5 principaux partis politiques français pour le corpus *Polop*.

Le Tableau 5.5 présente les caractéristiques des 2 corpus après prétraitement.

Table 5.5: Volumétrie des différents corpus

Corpus	Nb de classes	Nb de documents	Nb de descripteurs	Nb de descripteurs distincts
Reuters	39	14 701	1 237 264	59 281
Polop	5	1 186	1 579 374	16 593

Nous observons un comportement similaire de nos approches pour ces deux corpus. Bien que proche des meilleurs algorithmes de comparaison, notre approche n'est pas plus performante en Micro F-mesure, ni en Macro F-mesure (voir Figure 5.2 pour le corpus *Polop*).

3. `http://www.lirmm.fr/~bouillot/polop`

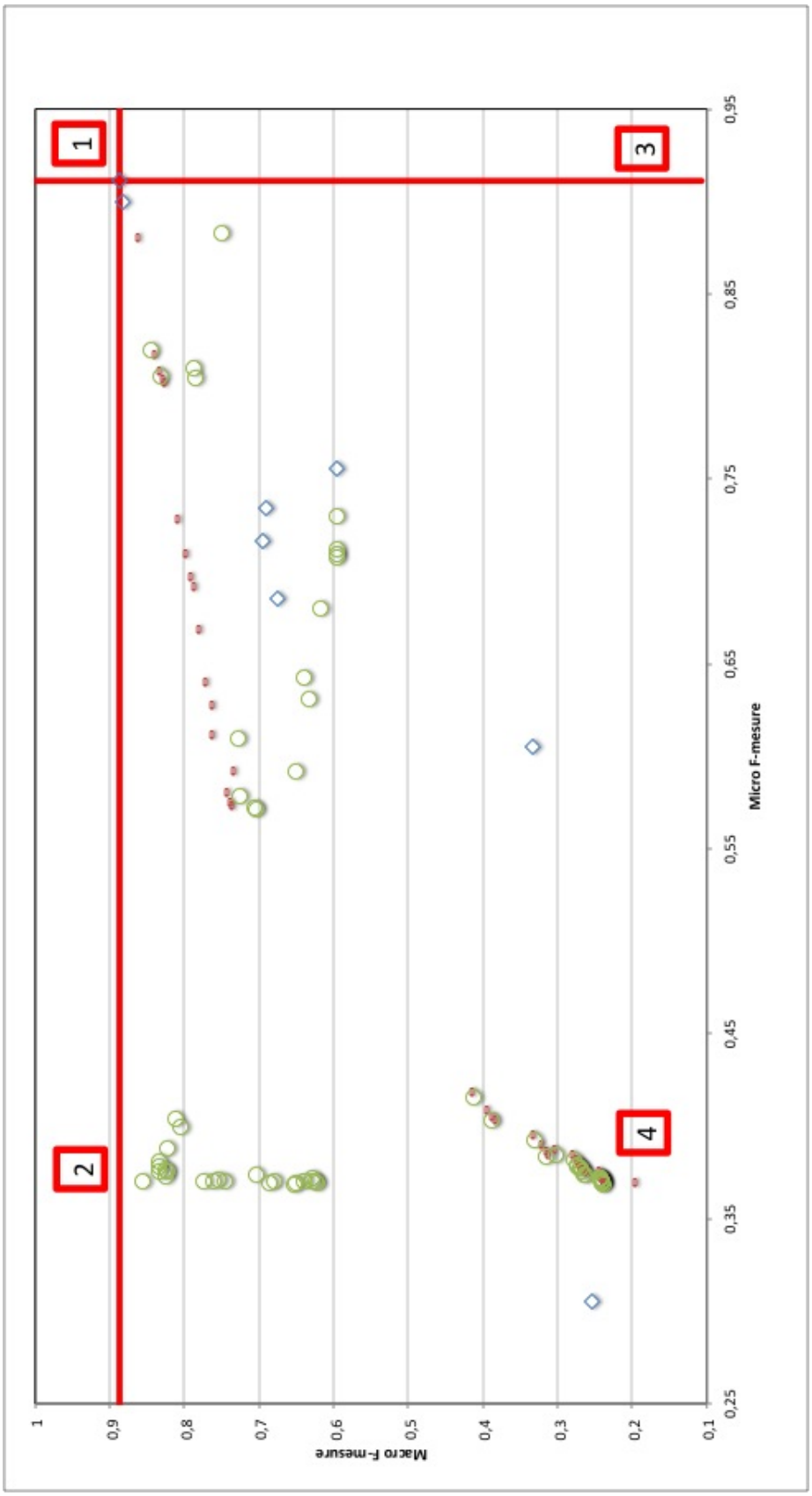


Figure 5.2: Résultats Corpus *Polop*

Un résumé des algorithmes les plus performants est donné dans le Tableau 5.6 pour le corpus *Polop*.

Table 5.6: Résultats détaillés obtenus sur le corpus *Polop*

Algorithme	α	β	ω	MicroF-mesure	MacroF-mesure	Moyenne Micro et Macro
SMO				0.911	0.887	0.899
DMNB				0.900	0.884	0.892
CFC	0	1	0	0.879	0.864	0.871
Naive Bayes	0.25	1	0	0.820	0.844	0.832
CFC	0.25	1	0	0.815	0.840	0.828
CFC	0.5	1	0	0.806	0.833	0.820
Naive Bayes	0.75	1	0	0.805	0.832	0.819
Naive Bayes	0	1	0	0.883	0.751	0.817
CFC	0.75	1	0	0.802	0.830	0.816
CFC	1	1	0	0.800	0.828	0.814
Naive Bayes	0.5	1	0	0.809	0.789	0.799
Naive Bayes	1	1	0	0.804	0.786	0.795
CFC	0	0.75	0	0.726	0.810	0.768
CFC	0	0.5	0	0.707	0.798	0.753
CFC	0	0.25	0	0.695	0.792	0.743
CFC	0	0	0	0.690	0.789	0.739
CFC	0	0	0.25	0.667	0.782	0.724
NaiveBayes				0.735	0.691	0.713
LadTree				0.716	0.696	0.706
j48				0.685	0.676	0.681
ComplementNaiveBayes				0.756	0.597	0.676
Naive Bayes	0	0.75	0.25	0.610	0.728	0.669
Naive Bayes	0	0.75	0	0.730	0.597	0.664
Naive Bayes	0	0.5	0	0.712	0.595	0.653
Naive Bayes	0	0.25	0	0.709	0.596	0.653
Naive Bayes	0	0	0	0.707	0.597	0.652
NaiveBayesMultinomial				0.605	0.332	0.469
LibSVM				0.195	0.107	0.151

Cette seconde expérimentation de résultats permet de compléter nos observations précédentes.

1. Selon le corpus, les paramètres prépondérants peuvent varier. Si α semblait être important pour le corpus précédent, β semble l'être sur le corpus *Polop*. Une valeur de β élevée permet d'obtenir de meilleurs résultats sur le corpus *Polop* indépendamment de la valeur de α.
2. Nos propositions fonctionnent bien, mais légèrement moins bien que les algorithmes SMO et DMNB
3. L'approche CFC permet d'obtenir de meilleurs résultats que l'approche Naive Bayes sur ce corpus.
4. Les paramètres qui donnent les meilleurs résultats avec l'approche Naive Bayes sont aussi les meilleurs avec l'approche CFC.

Afin de mieux appréhender le comportement de nos pondérations et l'impact des paramètres, nous avons effectué quatre séries d'expérimentations visant à étudier notre approche en fonction des caractéristiques d'un corpus. Un corpus étant constitué de classes, de documents et de descripteurs, nous avons voulu étudier :

1. L'impact du nombre de classes ;

2. L'impact du nombre de documents ;
3. L'impact du nombre de descripteurs ;
4. L'impact du déséquilibre entre les classes.

Pour cela, nous avons décidé de choisir un corpus, de relever une première série de mesures puis d'appliquer des modifications (par exemple en supprimant des classes) avant de relever une nouvelle série de mesures puis de modifier encore le corpus, etc. En plus de pouvoir comparer notre approche aux autres algorithmes, notre objectif est d'étudier le comportement des différentes solutions en fonction de l'évolution du corpus. Nous nous sommes tout d'abord intéressés à l'impact du nombre de classes sur les autres algorithmes ainsi que sur nos propositions.

5.5.1 Conséquences du nombre de classes sur la classification

Nous nous intéressons tout d'abord à l'impact du nombre de classes sur le corpus *"Reuters"*. Pour cela nous avons fixé aléatoirement le nombre de documents par classe (50) puis nous avons supprimé des classes pour passer de 28 à 2 classes. Nous avons réalisé 7 expérimentations dont les résultats sont présentés dans le tableau 5.7.

Par souci de clarté et de lisibilité, nous avons ainsi choisi de ne présenter qu'une vision partielle des résultats. Nous avons choisi de faire figurer :
- Le résultat des 8 algorithmes de comparaison ;
- Les 3 meilleures combinaisons de paramètres obtenues pour la première expérimentation avec l'approche Naive Bayes ;
- Les 3 meilleures combinaisons de paramètres obtenus pour la dernière expérimentation avec l'approche Naive Bayes ;
- Les 3 meilleures combinaisons de paramètres obtenues pour la première expérimentation avec l'approche CFC ;
- Les 3 meilleures combinaisons de paramètres obtenues pour la dernière expérimentation avec l'approche CFC.

Lorsqu'une combinaison de paramètres était à la fois dans le top 3 de la première et de la dernière expérimentation, nous avons décidé de ne pas ajouter d'autres résultats. Ainsi pour chaque série, nous affichons entre 14 (8 pour les algorithmes de comparaison, 3 pour l'approche Naive Bayes et 3 pour l'approche CFC) et 20 résultats (8 pour les algorithmes de comparaison, 6 pour l'approche Naive Bayes et 6 pour l'approche CFC). Nous décidons d'afficher les résultats obtenus en considérant la moyenne des F-mesures Micro et Macro. Enfin, pour faciliter la lecture nous indiquons en rouge-gras les maximums obtenus pour une série et une famille d'algorithme donnée pour chaque série (le maximum pour les algorithmes de comparaison, le maximum pour l'approche CFC et le maximum pour l'approche Naive Bayes).

La première partie du tableau comporte des informations relatives aux jeux de données. Par exemple, dans le tableau 5.7, nous pouvons voir dans la première partie du tableau le nombre de classes diminuer (ligne Classes), entrainant de facto une baisse du nombre de documents (ligne Doc) mais pas du nombre de documents moyen par classe (ligne Doc Moyen). Dans la seconde, nous retrouvons les résultats obtenus. Par exemple, pour la première série, avec 28 classes, le résultat moyen de la Micro et Macro F-mesure obtenu avec l'approche CFC (première ligne non colorée) et les paramètres ($\alpha = 0.25$, $\beta = 1$ et $\omega = 0$) est de 0.710. Le fait qu'il soit écrit en rouge-gras signale que 0.710 est le maximum obtenu pour l'approche CFC (quels que soient les paramètres utilisés) pour la série 1.

Table 5.7: Impact de la diminution du nombre de classes sur la moyenne des Micro et Macro F-Mesure

				Datasets						
Classes				28	25	20	15	10	5	2
Doc				1400	1250	1000	750	500	250	100
Doc Moyen				50	50	50	50	50	50	50
Descripteurs				164 540	148 693	124 736	97 517	62 808	31 101	10 508
				Résultats						
Algorithme	α	β	ω	1	2	3	4	5	6	7
ComplementNaiveBayes				**0.727**	**0.750**	**0.744**	**0.769**	**0.824**	0.927	0.971
DMNB				0.649	0.684	0.677	0.713	0.781	**0.929**	**0.980**
j48				0.619	0.657	0.681	0.725	0.796	0.909	0.971
LadTree				0.311	0.319	0.420	0.550	0.815	0.887	0.971
LibSVM				0.721	0.727	0.725	0.752	0.790	0.873	0.959
NaiveBayes				0.525	0.543	0.554	0.598	0.669	0.849	0.954
NaiveBayesMultinomial				0.693	0.712	0.709	0.745	0.791	0.905	0.971
SMO				0.655	0.674	0.676	0.686	0.766	0.929	0.980
CFC	0.25	1	0	**0.710**	**0.727**	0.726	**0.759**	0.820	**0.941**	0.923
CFC	0.25	0.75	0.25	0.706	0.726	**0.731**	0.754	**0.823**	0.937	0.961
CFC	0.5	1	0	0.704	0.725	0.728	**0.759**	0.818	0.937	0.920
CFC	0	0.333	0.333	0.672	0.701	0.694	0.721	0.801	0.935	**0.989**
CFC	0	0	0.75	0.667	0.692	0.694	0.710	0.801	0.923	0.979
CFC	0	0	0.5	0.657	0.685	0.688	0.704	0.796	0.925	0.979
Naive Bayes	0.25	0.75	0.25	**0.707**	0.723	**0.730**	0.755	**0.827**	0.937	0.961
Naive Bayes	0.25	1	0	**0.707**	0.730	0.725	0.759	0.813	0.933	0.941
Naive Bayes	0	0.75	0.25	0.703	**0.731**	0.724	**0.761**	**0.827**	**0.943**	0.950
Naive Bayes	0	0.5	0.5	0.697	0.722	0.714	0.746	0.817	**0.943**	**0.980**
Naive Bayes	0	0	0.25	0.648	0.676	0.680	0.707	0.800	0.913	0.979
Naive Bayes	0	0	0	0.638	0.665	0.675	0.703	0.800	0.909	0.979

Comme nous pouvions le supposer, l'ensemble des algorithmes ont un meilleur comportement en présence d'un nombre limité de classes. Les algorithmes, quels qu'ils soient obtiennent de meilleurs résultats lorsque le nombre de classes diminue.

Nous pouvons remarquer que les approches fondées sur les arbres de décision sont plus impactées que les autres par le nombre de classes. Nos propositions utilisées avec les meilleurs paramètres donnent des résultats comparables avec les autres approches. Ils sont légèrement en dessous des meilleurs sur la première série (mais plus performants que la plupart tout de même) et légèrement au-dessus sur la dernière. Pour ce corpus les meilleures combinaisons de paramètres semblent être obtenues avec une valeur élevée pour α. En revanche pour les paramètres β et ω, nous pouvons remarquer que :

- avec un nombre de classes plus important, il vaut mieux privilégier un β tendant vers 1.
- avec un nombre de classes faible, des valeurs équilibrées donnent des résultats plus élevés.

Nous nous sommes intéressés par la suite à l'impact du nombre de documents.

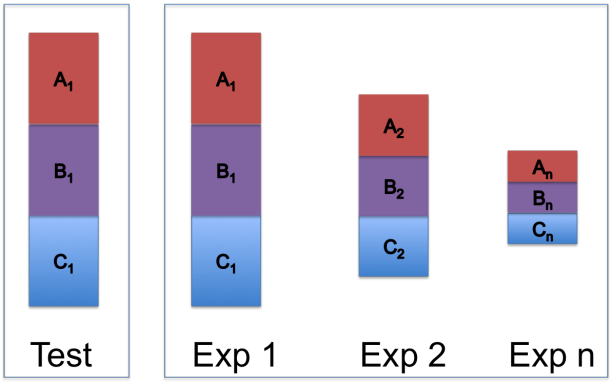


Figure 5.3: Processus de validation

5.5.2 Conséquences du nombre de documents sur la classification

Dans cette deuxième série d'expérimentations, nous nous concentrons sur l'étude de l'impact du nombre de documents par classes. Pour cela, nous avons fixé 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 5.8.
Pour ces expérimentations, nous avons utilisé un processus de validation croisée que nous avons adapté afin que le nombre de documents disponibles pour le test ne diminue pas en même temps que le nombre de documents disponibles en apprentissage (un nombre trop faible de documents en phase de test biaise les mesures relevées). Pour cela, nous avons utilisé lors de la première série une validation croisée classique (A_1, B_1, C_1) puis nous avons conservé cette répartition des documents de test (A_1, B_1, C_1) pour les autres séries comme illustré dans la Figure 5.3.
Ainsi les documents de test, leur nombre et leur répartition, sont identiques pour l'ensemble des séries.

Table 5.8: Impact de la diminution du nombre de documents par classe sur la moyenne des Micro et Macro F-Mesure

Datasets

Classes				10	10	10	10	10	10	10	10	10
Doc				500	450	390	330	270	210	150	90	30
Descripteurs				62 808	57 336	47 753	42 219	33 572	26 040	17 596	9 641	3 023
Descripteurs Moyens				126	127	165	128	124	124	117	107	101
Descripteurs distincts				9 629	9 112	8 491	7 857	6 929	5 776	4 610	3 161	1 414

Resultats

Algorithme	α	β	ω	1	2	3	4	5	6	7	8	9
ComplementNaiveBayes				**0.807**	0.793	0.781	0.783	0.769	0.751	0.724	**0.685**	**0.518**
DMNB				0.645	0.759	0.756	0.737	0.710	0.562	0.533	0.536	0.376
j48				0.795	0.796	0.796	0.798	**0.795**	**0.771**	0.767	0.588	0.091
LadTree				0.796	**0.805**	**0.810**	**0.802**	0.789	0.763	**0.777**	0.508	0.162
LibSVM				0.695	0.713	0.657	0.586	0.537	0.464	0.440	0.298	0.206
NaiveBayes				0.636	0.622	0.611	0.582	0.555	0.543	0.514	0.429	0.268
NaiveBayesMultinomial				0.736	0.772	0.767	0.761	0.745	0.731	0.704	0.676	**0.518**
SMO				0.721	0.722	0.708	0.681	0.644	0.604	0.570	0.409	0.223
CFC	0.25	0.5	0.5	**0.825**	0.797	0.785	0.766	0.776	**0.765**	**0.723**	0.633	0.428
CFC	0.25	0.75	0	0.824	0.797	0.782	0.760	0.774	0.762	0.724	0.634	0.429
CFC	0.25	0.75	0.25	0.823	**0.800**	**0.788**	**0.772**	0.780	0.763	0.728	0.633	0.428
CFC	0.25	0	0.75	0.821	0.794	0.776	0.757	0.772	0.762	0.720	0.627	0.429
CFC	0	0.75	0.25	0.774	0.786	0.782	0.768	0.780	0.756	0.714	0.642	**0.467**
CFC	0	1	0	0.772	0.779	0.776	0.766	**0.782**	0.755	0.719	**0.646**	**0.467**
CFC	0	0.5	0.5	0.768	0.782	0.771	0.760	0.776	0.754	0.709	0.641	0.467
Naive Bayes	0.25	0.75	0.25	0.827	**0.794**	**0.782**	0.766	0.763	0.749	**0.714**	0.611	0.421
Naive Bayes	0.25	0.75	0	**0.823**	0.792	0.779	0.756	0.762	**0.755**	0.712	0.612	0.421
Naive Bayes	0	0.75	0.25	0.775	0.785	0.783	0.767	**0.772**	0.749	0.709	**0.645**	**0.497**
Naive Bayes	0	1	0	0.775	0.783	0.779	**0.769**	**0.772**	0.750	0.713	0.645	**0.497**
Naive Bayes	0	0.25	0.75	0.772	**0.782**	0.771	0.753	0.766	0.749	0.702	0.642	**0.497**

Conformément à ce que nous pouvions attendre, les résultats diminuent lorsque le nombre de documents disponibles en apprentissage diminue puisqu'un faible nombre de documents ne permet pas de construire un modèle aussi fiable que ceux construits à partir d'un grand nombre de documents. Néanmoins, il est intéressant de constater que l'impact est différent selon les catégories d'algorithmes. Par exemple, un très faible nombre de documents impacte fortement les approches de type SVM (LibSVM ou SMO) et les approches de type arbres de décision (j48 et LadTree) alors que les classifieurs bayésiens sont moins impactés. Cela s'explique par les choix imposés par la construction des modèles, par exemple, de façon schématique, les approches de type SVM positionnent des points dans l'espace (les points étant des documents) et vont chercher à déterminer des zones qui appartiendraient à telle ou telle classes. Plus le nombre de points dans l'espace est faible, plus il sera difficile de déterminer correctement les zones. Concernant nos propositions, nous pouvons noter qu'aux mêmes titres que les classifieurs bayésiens, elles résistent assez bien à une diminution du nombre d'exemples disponibles pour créer le modèle. Nous obtenons là aussi des résultats similaires aux algorithmes de comparaison (parfois légèrement meilleurs, parfois légèrement moins performants). Concernant les paramètres, retrouver une valeur faible pour α qui minimise ainsi l'impact de la pondération intra-classe basée sur les documents est logique puisque plus le nombre de documents sera faible et moins cette pondération sera précise.

Ensuite, nous nous sommes intéressés à l'impact du nombre de descripteurs sur la classification.

5.5.3 Conséquences du nombre de descripteurs sur la classification

Afin d'utiliser un corpus ayant un ration nombre de documents - nombre de mots qui soit suffisamment important, nous avons utilisé le corpus *Polop*. Nous avons fixé le nombre de classes (5) et de documents (1186) et décidé de supprimer aléatoirement des descripteurs afin de diminuer le nombre de descripteurs par document. Nous avons réalisé onze expérimentations résumées dans le Tableau 5.9. Sur les dernières séries, le nombre de documents diminue, car certains se sont retrouvés entièrement vidé de leur contenu.

Table 5.9: Impact de la diminution du nombre de descripteurs sur la moyenne des Micro et Macro F-Mesure

Datasets

Classes				5	5	5	5	5	5	5	5	5	5	5
Doc				1 186	1 186	1 186	1 186	1 186	1 186	1 186	1 172	1 168	1 156	1 137
Descripteurs				1 579 374	1 322 148	613 777	264 025	202 166	157 177	76 851	60 630	47 573	35 809	24 791
Descripteurs Moyens				1 332	1 115	518	223	175	133	66	52	41	31	22
Descripteurs Distincts				16 593	15 993	13 441	10 633	9 803	9 029	7 007	6 634	6 175	5 598	4 894

Résultats

Algorithme	α	β	ω	1	2	3	4	5	6	7	8	9	10	11
ComplementNaiveBayes				0.676	0.665	0.654	0.624	0.549	0.481	0.412	0.407	**0.390**	**0.368**	0.325
DMNB				0.892	**0.915**	**0.874**	**0.752**	**0.662**	**0.540**	**0.441**	0.414	0.351	0.310	0.289
j48				0.681	0.670	0.647	0.478	0.409	0.350	0.317	0.318	0.312	0.306	0.313
LadTree				0.706	0.709	0.648	0.509	0.479	0.429	0.360	0.340	0.317	0.302	0.301
LibSVM				0.151	0.186	0.245	0.435	0.434	0.433	0.376	0.378	0.367	0.363	**0.328**
NaiveBayes				0.713	0.679	0.594	0.491	0.435	0.398	0.366	0.353	0.332	0.344	0.289
NaiveBayesMultinomial				0.469	0.466	0.453	0.418	0.380	0.426	0.404	0.420	0.374	0.360	0.316
SMO				**0.899**	0.884	0.800	0.670	0.559	0.452	0.417	0.409	0.379	0.363	0.321
CFC	1	0.75	0	0.307	0.333	0.335	0.316	0.354	0.359	0.352	0.374	0.372	0.360	0.378
CFC	1	1	0	0.814	0.811	0.724	0.582	0.525	0.429	0.434	0.444	0.401	0.385	0.377
CFC	0.5	0.75	0	0.307	0.335	0.339	0.332	0.357	0.370	0.367	0.383	0.365	0.366	**0.379**
CFC	0.5	1	0	0.816	0.811	0.725	0.590	0.543	0.442	0.444	0.448	**0.417**	0.387	0.377
CFC	0	1	0	**0.820**	**0.817**	**0.728**	**0.602**	**0.546**	**0.457**	**0.465**	**0.455**	0.413	**0.398**	0.364
Naive Bayes	1	1	0	0.828	0.821	0.751	0.634	0.576	0.498	0.486	0.473	**0.439**	**0.395**	0.354
Naive Bayes	0.5	1	0	**0.871**	**0.850**	**0.813**	**0.697**	**0.648**	**0.535**	**0.497**	**0.489**	0.430	**0.403**	0.349
Naive Bayes	0	0.333	0.333	0.572	0.316	0.321	0.324	0.336	0.341	0.352	0.360	0.370	0.360	0.372
Naive Bayes	0	0.5	0	0.505	0.305	0.310	0.318	0.339	0.337	0.351	0.363	0.370	0.363	0.373
Naive Bayes	0	1	0	0.795	0.809	0.727	0.580	0.530	0.437	0.433	0.435	0.400	0.384	**0.385**

Comme nous pouvions nous en douter, les résultats diminuent au fil des expérimentations. Moins il y a de descripteurs et plus il est difficile de construire un modèle fiable. De plus, il n'est pas exclu que la suppression aléatoire de descripteurs dans les documents ait introduit du bruit en changeant la nature des documents. Nous pourrions ainsi retrouver des documents sans lien avec la classe auquel ils sont rattachés et les utiliser pour construire les modèles. Comme précédemment nous pouvons constater que nos propositions permettent d'obtenir de performances proches voir supérieures aux algorithmes de comparaison. Elles permettent d'obtenir de meilleures performances lorsque le nombre de descripteurs diminue bien que l'ensemble des performances soit globalement fortement impacté par un faible nombre de descripteurs.

Pour finir, lors de la dernière série, nous nous sommes intéressés à l'étude du déséquilibre entre classes.

5.5.4 Conséquences du déséquilibre entre classes sur la classification

Il convient tout d'abord de discuter de la notion de "déséquilibre". Un corpus peut être constitué de classes possédant toutes le même nombre de documents, on dit alors que le corpus est composé de classes équilibrées. A contrario, il est possible que certaines classes possèdent un nombre bien plus important de documents que d'autres classes et nous dirons alors que le corpus est composé de classes déséquilibrées. Il est très rare que des corpus soient équilibrés de façon naturelle et généralement, plutôt que de définir un corpus comme étant équilibré ou déséquilibré, nous allons considérer le corpus comme étant plus ou moins déséquilibré. Nous mesurons le déséquilibre d'un corpus en calculant le rapport entre la classe comportant le plus de documents et celle comportant le moins de documents. Il existe une différence selon que ce rapport soit de 2 (la classe la plus volumineuse contient deux fois plus de documents que la classe la moins volumineuse) ou de 1000.
Pour étudier l'impact du déséquilibre, nous avons sélectionné aléatoirement 6 classes déséquilibrées sur le corpus *Reuters* puis nous avons diminué le déséquilibre en supprimant des documents des classes majoritaires jusqu'à obtenir des classes équilibrées. Comme nous restons sur une problématique liée aux petits volumes, le déséquilibre étudié reste limité. Les résultats des 6 itérations réalisées sont présentés dans le tableau 5.10.

Pour compléter l'évaluation du déséquilibre (Noté D-Doc) et pour ne pas nous limiter au minimum et maximum, nous mesurons aussi l'écart type (noté ET-Doc) du nombre de documents. Nous indiquons aussi, pour information, le déséquilibre et l'écart type du nombre de descripteurs (notés D-Descripteurs et ET-Descripteurs).

Table 5.10: Impact de la diminution du déséquilibre entre classes sur la moyenne des Micro et Macro F-Mesure

Datasets

Classes				6	6	6	6	6	6	6	6
Doc				2 164	2 047	1 865	1 587	1 287	904	504	300
Descripteurs				214 147	201 443	183 297	158 391	127 381	89 427	52 293	31 230
D-Doc				14	12	10	8	6	4	2	1
ET-Doc				254	229	198	156	115	70	23	0
D-Descripteurs				12	10	9	7	5	4	2	1
ET-Descripteurs				24 922	22 160	18 871	15 061	10 889	6 484	2 138	681

Résultats

Algorithme	α	β	ω	1	2	3	4	5	6	7	8
ComplementNaiveBayes				0.557	0.571	0.570	0.568	0.586	0.600	0.602	0.571
DMNB				0.554	0.558	0.557	0.571	0.580	0.603	0.609	0.597
j48				0.518	0.531	0.553	0.567	0.594	0.584	0.562	0.571
LadTree				**0.592**	**0.595**	**0.640**	**0.640**	**0.650**	**0.659**	**0.642**	**0.598**
LibSVM				0.555	0.569	0.574	0.588	0.608	0.620	0.606	0.567
NaiveBayes				0.517	0.523	0.530	0.537	0.533	0.543	0.546	0.531
NaiveBayesMultinomial				0.488	0.485	0.492	0.478	0.506	0.559	0.580	0.557
SMO				0.509	0.514	0.525	0.543	0.570	0.594	0.600	0.590
cfc1	0.75	0.75	0.25	0.630	0.621	0.605	0.579	0.563	0.557	0.556	0.525
cfc1	0.5	0.5	0.5	0.620	0.607	0.595	0.564	0.553	0.551	0.553	0.526
cfc1	0.5	0.75	0	0.613	0.605	0.597	0.571	0.556	0.559	0.564	**0.527**
cfc1	0.25	0.75	0.25	0.636	**0.630**	0.617	0.585	0.572	0.568	0.562	0.522
cfc1	0.25	1	0	**0.639**	**0.630**	0.619	0.592	**0.583**	0.574	0.567	0.524
cfc1	0	1	0	0.637	0.629	**0.623**	**0.598**	**0.583**	**0.584**	**0.570**	0.508
Naive Bayes	0.5	0.75	0.25	0.629	0.632	0.631	0.624	0.623	0.612	0.590	0.573
Naive Bayes	0.5	1	0	0.641	0.648	0.644	0.634	0.635	0.623	0.596	0.572
Naive Bayes	0.25	1	0	**0.647**	**0.649**	0.648	0.642	0.646	0.634	0.594	**0.575**
Naive Bayes	0	1	0	0.644	**0.649**	**0.649**	**0.644**	**0.649**	**0.647**	**0.607**	0.574

Il est difficile de tirer des conclusions de cette expérimentation, car en "jouant" sur le déséquilibre, nous impactons aussi la qualité globale du modèle en diminuant le nombre de documents disponibles. Néanmoins, cette expérimentation met en évidence plusieurs éléments :

- Pour tous les algorithmes de comparaison, nous pouvons observer que les performances augmentent puis diminuent à mesure que les classes s'équilibrent. Nous pensons que l'augmentation des performances est liée à l'impact du déséquilibre (si un modèle est impacté par le déséquilibre alors il donne de moins bonnes performances lorsque le déséquilibre augmente), mais que la diminution sur les dernières itérations est liée au nombre de documents disponibles (nous avons étudié précédemment l'impact du nombre de documents sur le modèle).
- Nous pouvons observer un comportement similaire avec nos pondérations intégrées dans une approche NaiveBayes.
- Au contraire, nos propositions avec l'approche CFC n'augmentent pas sur les premières itérations ce qui implique que cette approche est plus appropriée pour traiter les corpus déséquilibrés.
- Parmi les algorithmes de comparaison, LadTree est particulièrement efficace sur ces corpus.

Nous synthétisons l'ensemble de nos observations dans la section suivante et nous discutons des avantages et des inconvénients de notre proposition.

Chapitre

6

Bilan et discussions

Nous avons présenté dans cette première contribution une méthode de pondération paramétrable que nous avons expérimentée sur des corpus différents. Nos propositions, avec les "bons" paramètres sélectionnés, permettent d'obtenir des résultats meilleurs que l'ensemble des algorithmes de comparaison sur le corpus *Itesoft* mais aussi dans près de 40% des autres corpus (et souvent proches des meilleurs le reste du temps) et ce indépendamment du corpus.
Pour illustrer nos propositions, nous avons réalisé un grand nombre de tests. 120 combinaisons de paramètres et 15 algorithmes de comparaisons ont été évalués sur 36 corpus différents. En validation croisée, cela représente plus de 14 000 évaluations réalisées.
Ces tests ont permis de valider l'intérêt de notre proposition pour traiter des corpus composés d'un "petit volume" de données :

1. Nos pondérations résistent mieux que les algorithmes de comparaison lorsque le nombre de descripteurs est faible.
2. Avec une approche CFC, elles sont plus résistantes sur les corpus déséquilibrés.
3. Elles sont plus adaptées lorsque le nombre de classes est faible et résistent aussi bien que les meilleurs algorithmes lorsque le nombre de documents est faible.
4. Le choix des paramètres α, β et ω a un impact significatif sur les résultats.
5. Le choix de l'approche retenue (CFC ou Naïve Bayes) peut avoir un impact significatif sur les résultats.

Table 6.1: Résumé des "meilleurs" paramètres pour l'approche Naïve Bayes

α	β	ω	Nombre de corpus
0	0.25	0.75	2
0	0.5	0.5	2
0	0.75	0.25	8
0	1	0	10
0.25	0.75	0	2
0.25	0.75	0.25	6
0.25	1	0	4
0.5	1	0	9
1	1	0	2

Table 6.2: Résumé des "meilleurs" paramètres pour l'approche CFC

α	β	ω	Nombre de corpus
0	0.33	0.33	1
0	0.5	0.5	1
0	0.75	0.25	1
0	1	0	16
0.25	0.5	0.5	3
0.25	0.75	0.25	7
0.25	1	0	7
0.5	0.75	0	2
0.5	1	0	2

Nous synthétisons dans les tableaux 6.1 et 6.2, les paramètres ayant permis d'obtenir les meilleurs résultats sur l'ensemble des 36 corpus utilisés jusqu'ici. Nous indiquons dans le dernière colonne le nombre de corpus concernés. Précisons que le nombre total est supérieur aux 36 corpus car pour un même corpus plusieurs ensembles de paramètres peuvent être retenus.

Les résultats obtenus lors de nos expérimentations nous permettent de dégager quelques tendances. Par exemple, dans 82% de nos expérimentations, les meilleurs résultats ont été obtenus avec une valeur de α faible (≤ 0.25). De même, dans 89% de nos expérimentations, les meilleurs résultats ont été obtenus avec une valeur de β élevée (≥ 0.75). Néanmoins, même si les paramètres $\alpha = 0$, $\beta = 1$ et $\omega = 0$ permettent d'obtenir de meilleurs résultats dans 30% des cas, nous pouvons remarquer qu'il n'existe pas un ensemble de paramètres qui permettent d'obtenir de meilleurs résultats indépendamment du corpus.

Les tests menés nous permettent aussi de tirer quelques enseignements sur les approches de la littérature :

1. Les classifieurs bayésiens sont efficaces pour traiter les problèmes de classification de textes. Les approches DMNB (8 fois), ComplementNaiveBayes (10 fois [1]) et dans une moindre mesure NaiveBayesMultinomial (2 fois [2]) ont permis d'obtenir de meilleurs résultats que les autres types d'approches dans 53% des cas.

1. dont 1 à égalité avec l'algorithme NaiveBayesMultinomial.
2. dont 1 à égalité avec l'algorithme ComplementNaiveBayes.

2. LadTree (12 fois) est aussi un candidat crédible.
3. En revanche, les approches de type SVM (LibSVM 1 fois et SMO 2 fois) ou l'algorithme j48 (2 fois) se sont montrés moins à leur avantage pour traiter les corpus utilisés dans les expérimentations.
4. Parmi les 8 algorithmes de comparaison retenus, seul l'algorithme Naive-Bayes n'a jamais été le plus performant sur un corpus donné.

Remarquons que même si le contexte applicatif et les exemples choisis pour illustrer notre proposition sont mono classes (un document appartient à une seule classe), nos propositions sont tout à fait adaptées pour être utilisées dans un contexte multi-classes en considérant simplement un document appartenant à plusieurs classes comme plusieurs documents distincts appartenant à plusieurs classes distinctes.

Dans cette partie, nous avons étudié l'impact de la composition du corpus sur les paramètres, mais aussi sur un ensemble d'algorithmes établis et éprouvés et nous sommes arrivés à la conclusion que :
- Il n'existe pas d'algorithmes plus performants que les autres indépendamment du corpus ;
- Il n'existe pas de paramètres plus performants que les autres indépendamment du corpus.

Ces éléments nous ramènent à un constat établi par le théorème *No Free Lunch* [183] qui sera présenté dans la seconde partie de ce manuscrit. Pour définir le meilleur couple approche-paramètres, nous avons testé empiriquement une partie de l'ensemble des possibles (faut-il utiliser une approche CFC ou une approche NaiveBayes ? Quelles valeurs faut il choisir pour α, β et ω ?). Cette méthode pour détecter la meilleure combinaison, bien qu'efficace en terme de résultat, est aussi coûteuse en temps qu'inélégante dans sa construction. C'est pourquoi nous nous interrogeons dans la seconde partie de ce manuscrit sur les solutions existantes pour traiter ce type de problématique et nous proposons une approche originale de la traiter.

Deuxième partie

De nouveaux méta-descripteurs pour représenter un corpus

Chapitre

7

Introduction

Dans la première partie de ce manuscrit, nous avons défini un ensemble de couples algorithmes-paramètres et prouvé leur efficacité sur des corpus différents. Les expérimentations menées montrent que :

- Il n'existe pas d'algorithmes plus performants que les autres indépendamment du corpus ;
- Il n'existe pas de paramètres plus performants que les autres indépendamment du corpus.

Cela rejoint les conclusions présentées dans [182] pour qui la définition d'un "bon" algorithme est une tâche qui prend du temps et implique de mener un grand nombre d'expérimentations sur un grand nombre de classifieurs et de comparer les performances des uns et des autres. Les mêmes auteurs ont défini et formalisé le théorème du *No Free Lunch* dans [183] qui stipule que si un classifieur A superforme un classifieur B pour un ensemble de cas donnés, il existe autant de cas où B superforme A. En d'autres termes, il n'est pas possible de construire un classifieur qui fonctionne mieux que les autres indépendamment du problème donné [1, 139, 148, 177]. Les motivations théoriques du théorème du *No Free Lunch* sont données dans [183] et [140].

Devant la grande diversité d'algorithmes pouvant être considérés comme candidats crédibles pour répondre à un problème de classification donné, une des grandes difficultés du domaine est de prédire quel sera le meilleur [66]. La sélection du meilleur classifieur est une tâche coûteuse [105] et est difficile parce que les fondements des algorithmes sont différents [181]. Pour [19], chaque algorithme a un *"Selective priority"*, c'est-à-dire qu'il est meilleur pour traiter un type de

problème spécifique, car chaque algorithme possède un *"inductive bias"* [103]. Des exemples d*"inductive bias"* sont donnés dans [15] :

- Les approches de type K-plus-proches-voisins fonctionnent mal si les données sont éparses (lié à la difficulté à trouver un voisin), mais au contraire fonctionnent bien si il y a beaucoup d'exemples.
- Les approches de type Naive Bayes sont robustes avec les données bruitées, manquantes ou rares et sont résistantes aux attributs non pertinents. En revanche les attributs corrélés dégradent les performances.
- Les approches de types SVM sont efficaces pour les problèmes d'optimisations et donnent de meilleurs résultats sur des données éparses.
- Les approches de types arbres traitent à la fois les données continues et discrètes et se comportent mieux sur les attributs nominaux ainsi qu'avec des données manquantes.

Des facteurs évidents tels que le volume de données (en relation avec la taille de l'espace de description), la proximité ("sparsity") des exemples ou encore le type de bruit influencent les classifieurs [176]. Le processus de sélection du modèle le plus approprié est décrit dans [20].

Depuis 20 ans, des travaux ont été menés afin de tenter de déterminer quel est le meilleur classifieur pour un problème de classification donné. La détection du meilleur classifieur peut être décomposé en deux tâches, la sélection de l'algorithme le plus performant, mais aussi la détection des paramètres optimaux pour un algorithme donné. En effet, de même qu'il n'existe pas de classifieur plus performant que les autres indépendamment du problème donné, il n'existe pas de paramètres optimums pour un algorithme indépendamment du problème donné [32, 112]. Ainsi si certains travaux se sont focalisés sur la détection du meilleur algorithme, d'autres se sont focalisés sur la détection des meilleurs paramètres [77, 71, 159, 4, 6, 5, 149, 112, 23, 75, 104, 48].

Il est cependant possible de regrouper les solutions proposées pour répondre à ces deux problématiques :

- Solution 1 : Tester l'ensemble des candidats possibles. C'est la solution qui fonctionne le mieux, mais est malheureusement impossible à appliquer, car il y a trop d'algorithmes candidats et trop de paramètres à considérer [27, 162].
- Solution 2 : Faire appel à un expert, qui pour un problème spécifique donné est capable de définir le meilleur classifieur. En supposant qu'il existe, cet expert peut être difficile à trouver [46].
- Solution 3 : Déterminer de façon automatique le ou les meilleurs candidats, solution qui concentre la majorité des travaux menés par la communauté.

Pour déterminer de façon automatique les candidats les plus probables, il existe deux grandes catégories.

- La première regroupe les approches qui consistent à converger vers le meilleur

classifieur (Problème d'optimisation) par itération successive (approche "*grid search*", programmation génétique) et est surtout utilisée pour la détection de paramètres [31]. L'idée générale de ces approches est d'exécuter plusieurs fois un classifieur en modifiant les paramètres utilisés afin de converger vers les plus performants. Les "*grid search*" restent les méthodes les plus utilisées tant que les ressources (temps, machine) le permettent ce qui est malheureusement impossible sur les ensembles trop volumineux pour lesquels des algorithmes génétiques seront plutôt utilisés [47].

- La seconde regroupe les approches qui consistent à utiliser les expériences passées pour prédire le futur [4, 153]. On parle alors de meta-learning ou méta-classification. L'idée générale est d'arriver à générer de la connaissance sur le lien entre les caractéristiques d'un problème de classification et les performances des classifieurs [157].

Il existe une grande différence entre les deux types d'approches concernant la durée nécessaire à l'obtention des meilleurs candidats. Avec les approches de méta-classification, la plus grande partie du temps nécessaire pour calculer les meilleurs candidats est portée en amont ; ceci est indépendant du problème de classification à résoudre. Ainsi, à partir du moment où le problème de classification est connu, le temps nécessaire pour déterminer les bons candidats sera moindre qu'avec les approches par itération. De plus, en ne prenant pas en compte les expériences passées, les approches par itération vont effectuer deux fois le même raisonnement pour arriver aux mêmes conclusions pour deux problèmes identiques. Enfin, les travaux récents suggèrent que la méta-classification est une solution fiable pour proposer un algorithme de classification qui s'appliquerait à l'ensemble des problèmes [18, 60, 144] et c'est l'approche retenue dans la suite de ce manuscrit.

Nous discutons dans le Chapitre 8 des notions liées à la méta-classification et nous effectuons une revue des principales solutions proposées dans la littérature. Dans le Chapitre 9, nous présentons une approche originale pour décrire un corpus et nous discutons des avantages de notre proposition par rapport aux approches de la littérature. Les expérimentations menées pour évaluer notre proposition sont décrites dans le Chapitre 10. Nous concluons cette seconde partie en discutant de notre proposition dans le Chapitre 11.

Chapitre

8 État de l'art de la méta-classification

Comme nous l'avons vu en introduction de cette seconde partie, la méta-classification vise à déterminer l'algorithme le plus approprié pour traiter un nouveau problème de classification par application des techniques de classification à partir des expériences passées [151]. Le terme méta-learning a été utilisé la première fois par [1] mais les approches de méta-classification trouvent leurs inspirations dans le problème de sélection d'un algorithme formalisé en 1976 dans [128] pour répondre à la question suivante : parmi tous les algorithmes disponibles, lequel est susceptible de donner les meilleurs résultats pour un problème spécifique ? L'auteur propose un modèle composé de 4 éléments.

- l'espace des problèmes, P, qui contient l'ensemble des données,
- l'espace des descripteurs, F, qui contient les caractéristiques mesurables obtenues sur P,
- l'espace des algorithmes, A, qui contient l'ensemble des algorithmes (classifieurs) pouvant résoudre le problème,
- l'espace des performances, Y, qui relie chaque algorithme de A avec des mesures de performances.

Toujours dans [128], le problème de sélection d'un algorithme est formalisé de la façon suivante : Pour toutes instances $x \in P$, à partir des descripteurs $f(x) \in F$, trouver $S(f(x)) = \alpha \in A$, tel que le classifieur sélectionné α maximise les performances $y(\alpha(x)) \in Y$ pour tout x. Ainsi, pour un nouveau problème de classification, il suffira d'appliquer l'algorithme α pour obtenir le meilleur résultat.

La difficulté est de déterminer la fonction S permettant d'obtenir le meilleur α. Aucune méthode pour résoudre le problème de sélection d'algorithmes n'est indiquée dans le papier et il n'est pas fait référence à la méta-classification, mais

il est spécifié que "la procédure de sélection est elle-même un algorithme", ce qui fait de la méta-classification une approche crédible pour résoudre le problème de sélection du meilleur algorithme [45].
Dans un contexte de méta-classification, nous sommes en présence d'un problème de sélection d'algorithmes. A est un ensemble de classifieurs et S est aussi un classifieur (on parle de méta-classifieur) qui peut éventuellement appartenir à A. P est l'ensemble des jeux de données, les problèmes étudiés par le passé et F les méta-descripteurs issus de P. À partir de ces définitions, la méta-classification peut être résumée ainsi :
À partir d'un ensemble de jeux de données (P), il est possible de construire une base de connaissance à partir de laquelle, par application d'un méta-classifieur (S), il sera possible de prédire les performances (Y) des différents algorithmes (A) pour un nouveau problème en utilisant à la fois les méta-descripteurs (F) du nouveau problème et les expériences passées (P).

Ces dernières années, les travaux de la communauté ce sont intéressés à ce problème et nous présentons dans la section suivante un bref état de l'art sur les méta-classifieurs S, les performances Y et les algorithmes A avant de voir plus en détail les problèmes P et les méta-descripteurs F.

8.1 Les algorithmes

Au cours des différents travaux, la méta-classification a fait l'objet de nombreuses applications. Par exemple l'utilisation de la méta-classification pour la détection du meilleur classifieur pour un dataset donné a été étudiée [10, 7] et pour prédire le classement entre algorithmes (ranking) dans [99, 17, 14, 154, 173]. Elle a aussi été utilisée pour prédire si un classifieur est capable de classer ou non (*Reliable classification*) dans [70]. Nous pouvons trouver d'autres applications comme la prédiction des performances (par exemple l'accuracy) pour un problème de classification donné dans [46, 172], la prédiction du taux d'erreur dans [44, 155, 13] ou encore la prédiction du temps de traitement dans [124]. L'utilisation de la méta-classification pour réduire le nombre de candidats possibles afin de déterminer les classifieurs à tester sans impacter la qualité de la classification obtenue (autrement dit sans éliminer les meilleurs algorithmes) est présentée dans [17, 83, 18]. La méta-classification a été utilisée pour prédire les paramètres d'un algorithme, par exemple pour sélectionner le type de noyau et les paramètres SVM [153, 4, 152, 102, 48] ou pour définir les paramètres d'un arbre de décisions dans [104]. Enfin, nous pouvons aussi citer son utilisation dans le cadre de la prédiction du *stopping point* (moment à partir duquel ajouter de la connaissance n'améliore pas les performances) dans [82] ou l'amélioration des performances des algorithmes génétiques dans [125] dans lequel les auteurs utilisent des techniques de méta-classification pour sélectionner un bon point de départ aux algorithmes

génétiques (les résultats obtenus dépendent du choix de la population de départ). Selon l'usage attendu, les mesures de performances (Y) ainsi que les algorithmes évalués (A) seront différents. Par exemple, si nous nous intéressons aux paramètres SVM permettant d'obtenir les meilleurs résultats ou bien que l'on étudie l'algorithme demandant le moins de ressources parmi l'ensemble des possibles, dans le premier cas A sera restreint aux approches de type SVM et Y à des mesures comme l'accuracy, la précision ou le rappel alors que dans le second A sera composé d'un ensemble d'algorithmes plus large emprunté aux différentes familles d'algorithmes (Naïve Bayes, SVM, Arbres de Décision...) et Y de mesures propres à la consommation de ressources (temps de traitement, mémoire utilisée...).

Ainsi, il convient d'abord de définir les algorithmes évalués (A). Tous les travaux relatés dans la littérature ont évalué un nombre variable d'algorithmes. L'espace de recherche (ou espace des hypothèses) définit l'ensemble des algorithmes (ou des modèles) possibles pouvant résoudre le problème donné. Le choix du mauvais algorithme peut être dû, entre autre, à l'absence du bon modèle dans l'espace de recherche. Il n'existe pas d'algorithme qui semble avoir les faveurs de la communauté conformément au théorème du No Free Lunch. Des techniques pour définir le nombre d'algorithmes à retenir ont été étudiées dans [18] mais ce choix n'a pas une grande influence sur les performances du système. Tout au plus peut-on signaler que certains auteurs ont utilisé un seul type d'algorithme [152, 102, 104] là où d'autres ont utilisé des algorithmes aux fondements différents [13, 166].
Nous donnons néanmoins un court aperçu des algorithmes utilisés dans la littérature pour illustrer la diversité des possibilités.

- Dans [13], les auteurs obtiennent de bons résultats de prédiction pour 8 algorithmes non paramétriques, 2 arbres de décision (LTree et C5.0Tree), Naïve Bayes, 2 méthodes par règles (Ripper et C5.0Rules), un classifieur discriminant linéairement, une méthode des plus-proches-voisin et une méthode combinant différentes approches (C5.0boost).
- Dans [100], les auteurs évaluent 20 algorithmes parmi lesquels des arbres de décisions, des réseaux de neurones, des algorithmes statistiques ou des méthodes par règles.
- Un arbre de décision (C5) et un SVM (LibSVM) sont utilisés dans [83].
- Des approches de type SVM dans [153, 4, 152, 102, 48].
- Dans [105], 8 classifieurs bayésiens sont évalués.
- Un arbre de décision (J48) dans [104].

Ces algorithmes sont évalués selon une ou plusieurs mesures de performances (Y).

8.2 Les performances

Nous présentons ici un aperçu des différents éléments relevés dans la littérature qu'il est possible de séparer en deux familles.

La première concerne des mesures de performances qualitatives. Il s'agit des travaux les plus anciens.

- La mesure de performances peut être simplement le meilleur algorithme [1, 68, 120] et dans ce cas le méta-classifieur va simplement déterminer quel est le meilleur algorithme pour un problème donné.
- De la même façon, dans [83], les auteurs se concentrent sur la prédiction du plus performant entre deux classifieurs, un arbre de décision (C5) et un SVM (LibSVM).
- Dans [100], pour chaque algorithme, une classe applicable ou non applicable est assignée selon que le taux d'erreur soit inférieur ou supérieur à un taux donné. Dans ce cas la mesure de performances est une valeur discrétisée pouvant prendre deux valeurs.

Cependant les travaux les plus récents utilisent plutôt des mesures quantitatives.

- Dans [85] et [123], les auteurs utilisent l'accuracy.
- [76] et [13] utilisent le taux d'erreur.
- [123] et [124] évaluent aussi les classifieurs selon le temps d'exécution.
- Dans [82], la mesure de performances correspond au *stopping point*.
- Dans [105], les classifieurs sont évalués avec 3 mesures de performances (*Accuracy*, *Mean Absolute Error* et *Relative Absolute Error*).
- Dans [15], 6 mesures de performances sont prises en compte (Précision, Rappel, Accuracy, ROC, ARR, Mean Absolute Error).
- Il est aussi possible d'évaluer les performances selon une combinaison de plusieurs critères (par exemple celui offrant le meilleur compromis entre ressources utilisées et taux de précision). Par exemple dans [69], la mesure de performances utilisée est une combinaison de l'accuracy, du temps d'apprentissage, du temps d'exécution et des ressources consommées.

On s'aperçoit que bien souvent l'accuracy est utilisée comme mesure de performances. L'idée générale est que si les modèles proposés sont fiables pour prédire l'accuracy, alors ils le seront aussi pour prédire le Rappel, la Précision ou toutes autres mesures de performances.

La méta-classification peut se décomposer en deux tâches, la prédiction d'une valeur qualitative et la prédiction d'une valeur quantitative. Ainsi, les méta-classifieurs (S) reposent soit sur des modèles de classification soit sur des modèles de régression [110].

8.3 Les méta-classifieurs

Dans la mesure où la méta-classification est un problème de classification, le théorème No Free Lunch s'applique aussi. Ainsi, il n'existe pas de méta-classifieur qui soit meilleur qu'un autre indépendamment du problème posé. C'est pourquoi nous

retrouvons dans la littérature différents types de classifieurs utilisés en tant que méta-classifieur. Par exemple [100] et [105] utilisent un arbre de décision alors que [13] utilise un modèle de régression linéaire. D'autres utilisent des approches combinant plusieurs classifieurs [78]. Enfin certains ont développé des approches spécifiques adaptées à leur problème [162, 158, 85]. Dans [53], les auteurs introduisent le méta-mining pour la recherche du meilleur méta-classifieur, les efforts ne portent plus sur la génération du modèle (méta-learning), mais sur les méta-classifieurs appliqués au modèle. Le méta-mining est aussi discuté dans [110].

Les méta-classifieurs S dépendent du contexte et de l'objectif, mais ils sont influencés par le nombre de jeux de données déjà étudié (P) pour créer le modèle. Nous discutons de ce point dans la section suivante.

8.4 Les problèmes

Comme dans tout problème de classification supervisée, la qualité du modèle final dépend du nombre d'exemples disponibles pour créer le modèle. Comme nous l'avons rappelé dans la première partie, les modèles sont plus performants lorsque le nombre d'exemples disponibles est important. Or les problèmes de méta-classification utilisent généralement un nombre limité de jeux de données. Par exemple, 22 jeux de données très variés sont utilisés dans [100] (données médicales, données bancaires, images), 101 dans [166] récupérés depuis le *UCI repository*. [104] utilise un ensemble de 14 jeux de données de l'éducation nationale utilisés pour la prédiction des performances des élèves. Dans [105], les expérimentations sont menées à partir de 39 jeux de données récupérées à partir du *UCI repository*, *Weka* et *Promise* et dans [123], les auteurs utilisent 83 jeux de données issus du *UCI repository*, *StatLib* et du livre Analyzing Categorical Data. La plupart des approches de la littérature sont testées sur moins de 100 jeux de données. Nous pouvons signaler toutefois que dans [162], les auteurs ont utilisé plus de 1000 jeux de données récupérés du *UCI repository*, *StatLib*, KDD et *Weka*. La difficulté principale est d'obtenir un nombre suffisamment important de jeux de données qui soit du même type que ceux sur lesquels auraient été calculées les performances (Y) pour un certain nombre d'algorithmes (A).
Chaque méta-exemple est un problème de classification étudié par le passé pour un jeu de données précis. Il représente l'ensemble des résultats (Y) des algorithmes testés (A) pour un problème donné (P). La génération de méta-exemples est difficile et certains auteurs se sont attaqués à ce problème. Par exemple [118] propose une méthode basée sur l'apprentissage actif et [16] propose une méthode pour réduire le nombre de méta-exemples nécessaire en ne sélectionnant que les plus pertinents. L'idée générale est de chercher les jeux de données les plus différents pour compléter les jeux de données déjà étudiés. La création artificielle de méta-exemples est étudiée dans [40] mais le coût pour générer un ensemble cohérent peut être trop important. Cela dépend notamment du nombre et de

la complexité des méta-classifieurs candidats, du type ou du volume de données [121, 119].

Pour autant, quelque soit leur nombre et leur type, chaque méta-exemple est constitué :

1. De mesures de performances (Y) observées pour un ou plusieurs algorithmes (A) ;
2. De méta-descripteurs (F) qui décrivent les propriétés des jeux de données (P) concernés par le problème de classification.

Dans les sections précédentes, nous avons discuté des jeux des données (P), des mesures de performances (Y), des méta-classifieurs (S) et des algorithmes (A). Il convient maintenant de discuter des méta-descripteurs (F).

8.5 Les méta-descripteurs

L'objectif de la méta-classification est de lier les performances (Y) d'un ensemble d'algorithmes (A) à un jeu de données (P) de façon à pouvoir prédire les performances d'un nouveau jeu de données par application d'un méta-classifieur (S) sur les expériences passées. Pour cela les caractéristiques des jeux de données (P) doivent être extraites de façon identique de manière à être comparables. Ces mêmes caractéristiques seront extraites des jeux de données dont on cherche à prédire les performances.
Une grande partie des travaux portant sur la méta-classification s'intéresse à la manière de décrire les jeux de données de façon la plus précise possible et plusieurs approches utilisant des fondements très différents ont été proposées.
Les premiers méta-descripteurs proposés sont des variables quantitatives calculées sur les jeux de données par exemple le nombre d'exemples d'apprentissage, le nombre de classes, etc [17, 37, 66].

Il est possible de regrouper ces méta-descripteurs quantitatifs en trois familles [76, 15] :

- Les descripteurs simples qui regroupent les propriétés accessibles rapidement : nombre de classes, nombre d'observations, nombre d'attributs ;
- Les descripteurs statistiques [37, 155] qui reposent sur des méthodes d'analyse statistique : moyenne, variance, matrice de covariance, coefficient d'aplatissement de Pearson, coefficient de dissymétrie ;
- Les descripteurs issus de la théorie de l'information [143] : Khi-2, information Gain et entropie des attributs et des classes.

Les approches empruntent généralement des méta-descripteurs issus des différentes familles. Par exemple, dans [16], les méta-descripteurs utilisés sont à la fois des mesures simples (le nombre d'attributs, le nombre d'attributs numériques, le

nombre d'attributs nominaux, le nombre de classes, le nombre d'instances incomplètes, le nombre d'instances) et des mesures statistiques (la moyenne, l'écart type, le coefficient d'aplatissement de Pearson, le coefficient de dissymétrie).
Dans [158] les méta-descripteurs utilisés correspondent au nombre d'attributs nominaux ou continus, au nombre de classes et d'exemples, au nombre d'attributs par classe, mais aussi aux distributions (distribution du nombre d'exemples au sein de la classe, distribution du nombre de données uniques, distribution du nombre de données manquantes). Dans [69], les auteurs utilisent en complément des histogrammes (histogramme de concentration, histogramme de corrélation).
Ces variables quantitatives sont parfois transformées en variables qualitatives. Par exemple, dans [104], les auteurs discrétisent 3 attributs numériques (le nombre d'attributs, le nombre de classes et le nombre d'instances) selon 3 valeurs possibles (Bas, Moyen, Haut). Dans [105], 3 méta-descripteurs discrétisés sont choisis, le nombre d'instances (inférieur ou égal à 286, supérieur à 286), le nombre d'attributs (supérieur à 16, inférieur ou égal à 16) et le pourcentage d'attributs nominaux (100%, entre 50 et 100%, inférieur à 50%).

Dans [11] et [114], les auteurs ont utilisé comme méta-descripteurs les propriétés d'un arbre de décision généré sur les jeux de données : ratio du nombre de nœuds de l'arbre sur le nombre d'attributs, ratio du nombre de nœuds de l'arbre sur le nombre d'instances en apprentissage, moyenne de la différence gain ratio, profondeur maximum de l'arbre entre la racine et la feuille la plus éloignée, nombre de nœuds répétés dans l'arbre, forme de l'arbre (Shape), homogénéité, nombre de feuilles divisées par la profondeur de l'arbre, nombre de sous-arbres identiques, etc.

Une autre approche a été proposée au début des années 2000 dans [115]. Il s'agit de l'approche des landmarkers. L'idée générale avancée par les auteurs est qu'il est possible d'utiliser les performances d'algorithmes pour caractériser un jeu de données. Le concept repose sur l'intuition que si deux jeux de données qui ont des caractéristiques similaires ont des résultats similaires pour un ensemble d'algorithmes donnés (ce qui est l'essence même de la méta-classification), alors deux jeux de données qui ont des résultats similaires pour en ensemble d'algorithmes donnés auront des caractéristiques identiques. Ainsi les auteurs proposent de mesurer les performances sur un ensemble restreint d'algorithmes pour définir les jeux de données et d'utiliser ces résultats comme méta-descripteurs. Les auteurs mettent en avant deux critères pour sélectionner les algorithmes les plus appropriés :

- *L'efficacité* : les algorithmes retenus doivent être le plus rapide possible ;
- *La diversité* : les algorithmes retenus doivent être issus de familles différentes, car les algorithmes construits sur les mêmes fondements auront des comportements et des biais similaires.

Le niveau de performances des algorithmes retenus importe peu puisqu'ils ne

servent qu'à caractériser le jeu de données. Par exemple, dans [115], les auteurs utilisent en plus de statistiques simples les accuracy obtenus via C5.0 trees, Naïve Bayes et une fonction linéaire discriminante.
Huit landmarkers sont utilisés dans [10] et [12] : One-level decision stump (meilleur nœud, moins bon nœud et choix aléatoire du nœud), un Naïve Bayes, 2 algorithmes des plus-proches-voisins, un classifieur linéaire et une classe par défaut. [125] utilisent quatre landmarkers (One-level decision stump (meilleur nœud, moins bon nœud et choix aléatoire du nœud) et Naïve Bayes) en complément de mesures statistiques et de mesures simples. D'après [83, 84], les approches de landmarking permettent de mieux prédire les performances des algorithmes que les méta-descripteurs statistiques.

Dans [43], les auteurs utilisent l'ordre entre les performances des landmarkers (*Relative Landmarking*) plutôt que les performances pures (approche Landmarking initiale). Cinq landmarkers sont utilisés : One-level decision stump (meilleur nœud, moins bon nœud et choix aléatoire du nœud), Naïve Bayes et une fonction discriminante linéaire.
[162] propose aussi des méta-descripteurs basés sur la relation entre algorithmes. Il suggère un nouveau modèle de génération de méta-descripteurs en comparant les algorithmes deux à deux et en proposant des règles du style "tel algorithme est meilleur que celui-ci". Il compare 20 algorithmes 2 à 2 (190 paires d'algorithmes) et considère ces 190 comparaisons comme méta-descripteurs en complément de 80 méta-descripteurs discutés précédemment (mesures simples, statistiques et landmarker).
Une autre approche qui utilise aussi les performances comme méta-descripteurs a été utilisée dans [83]. Les auteurs utilisent une approche basée sur les courbes d'apprentissage (learning curves) qui a montré son efficacité dans un contexte différent [82] pour détecter les *stopping point*. La méthode nécessite l'exécution d'expérimentations sur un petit nombre d'exemples puis sur un autre puis encore sur un autre... Ils recherchent la courbe d'apprentissage la plus proche. Les expérimentations montrent que la méthode donne de bons résultats, similaires aux autres approches. L'idée générale est que si le début de la courbe est proche (sur un sous-ensemble) alors la fin le sera aussi (sur l'ensemble des données).
Enfin, nous pouvons citer un dernier type de méta-descripteurs, les auteurs dans [124] utilisent le temps de calcul constaté pour la mesure des divers méta-descripteurs comme méta-descripteurs. Néanmoins cette approche nous parait trop dépendante de la machine utilisée pour les calculs pour être un méta-descripteur pertinent et pérenne.

Dans un contexte plus spécifique à la classification de documents, nous pouvons citer [78] et [41] qui utilisent des méta-descripteurs tels que le nombre d'exemples, le nombre de descripteurs moyen par document et par catégorie, le ratio entre le nombre d'exemples positifs et négatifs, la pondération moyenne des descripteurs par document par catégorie, la valeur maximale moyenne pour une catégorie, la

valeur minimale moyenne pour une catégorie, le nombre de descripteurs moyen ayant un poids supérieur à un seuil donné, le gain d'information moyen pour les meilleurs t descripteurs de chaque classe ou le nombre de descripteurs supérieurs à un gain d'information donné.

Les techniques de description d'un corpus décrites ci-dessus sont jugées comme pertinentes, mais certains méta-descripteurs peuvent être redondants ou peu pertinents [113]. Peu de travaux ce sont intéressés sur l'utilité et le pouvoir de discrimination de ces méta-descripteurs à part dans [69] et [168]. Ces derniers travaux se concentrent sur la sélection de méta-descripteurs dans un contexte de comparaison d'algorithmes de classification deux à deux. Ils poursuivent deux objectifs : (1) améliorer les performances et (2) comprendre les caractéristiques qui impactent sur les performances.
Les travaux cités précédemment visent à maximiser la capacité à prévoir sans vraiment chercher à comprendre les caractéristiques des *datasets* qui l'influencent. Le premier essai de sélection de méta-descripteurs est apparu dans [168]. Cependant les facteurs déterminants les performances pour un groupe d'algorithmes peut être différent pour un autre groupe. Dans [67], il est montré que des méta-descripteurs utiles pour prédire les performances d'un algorithme donné peuvent ne pas être utiles pour prédire les performances d'un autre algorithme.

Nous pensons qu'il existe une approche différente pour méta-décrire un corpus que nous présentons dans le chapitre suivant qui se base directement sur le corpus lui-même tout en apportant une vision plus précise que le nombre de classes ou de documents. Notre proposition s'appuie sur la définition de nouveaux méta-descripteurs.

Chapitre

9 De nouveaux méta-descripteurs

Nous pensons qu'il est possible de décrire un corpus en fonction de sa représentation dans l'espace. En positionnant les éléments du corpus, les uns par rapport aux autres, nous obtiendrons un schéma, une empreinte, qui définirait le corpus et nous pensons qu'il sera possible d'identifier deux corpus qui reproduiraient le même schéma. Par exemple la figure 9.1 représente le positionnement dans l'espace de 4 corpus chacun étant composé de 3 classes (cercles de couleur). Sur cet exemple, nous pouvons voir que les corpus 3 et 4 sont plus similaires qu'ils ne le sont avec les deux autres (qui sont eux-mêmes très différents l'un de l'autre).
Le problème revient alors à pouvoir projeter un corpus dans l'espace et de pouvoir positionner les différents éléments indépendamment du corpus. Pour cela nous proposons d'étudier la similarité de ses composants, les classes et les documents.

9.1 La similarité comme méta-descripteurs

Nous introduisons tout d'abord la notion de similarité entre composants.

Définition 9 (Similarité)

Deux composants (classes ou documents) sont similaires lorsque le vocabulaire utilisé dans les deux composants est similaire. Par extension, deux composants sont identiques lorsque le vocabulaire utilisé dans les deux composants est identique.

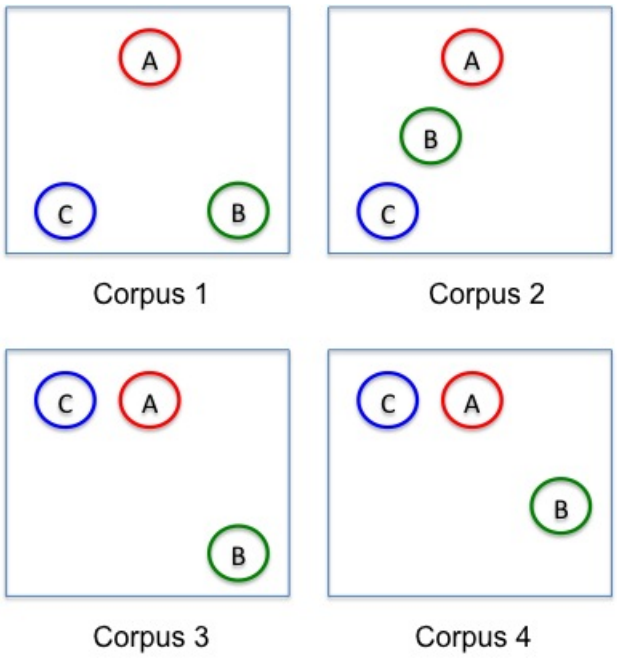


Figure 9.1: Corpus projeté dans l'espace

Tout d'abord, il est possible de mesurer la similarité entre deux classes, indépendamment des documents la composant. Intuitivement, les classes "chien" et "chat" (qui sont tous les deux des animaux de compagnie, mammifères à poil, vivant dans des maisons) seront plus similaires que les classes "chat" et "canard" (qui est un animal à plume, un oiseau qui pond des œufs et qui vit dans les mares). Un corpus peut être composé de classes similaires ou au contraire très différentes, mais il est tout à fait possible de trouver au sein d'un même corpus des classes similaires et des classes différentes. Nous introduisons la notion de similarité inter-classes.

Définition 10 (Similarité inter-classes)

Similarité inter-classes : similarité entre classes d'un corpus, indépendamment du nombre de documents ou de la répartition du vocabulaire au sein des documents.

De la même façon, il est possible de mesurer la similarité entre documents. Deux documents sont dits similaires lorsque les descripteurs utilisés dans ces documents sont similaires. Mesurer la similarité entre documents de classes différentes reviendrait in fine à étudier la similarité entre classes. En revanche, mesurer la similarité entre documents au sein de chaque classe offre une définition assez intéressante du corpus car les documents composant une classe peuvent être similaires les uns

des autres ou au contraire très différents les uns des autres. Par exemple, dans une classe regroupant des articles concernant "l'alimentation de la vache", les documents contiendront un vocabulaire assez répétitif et les documents seront assez similaires. À l'inverse et à volume de documents égal, une classe qui contiendrait des articles retraçant "la perception de la vache à travers le monde" auront un vocabulaire plus diffus et les documents, une similarité moins importante (le vocabulaire accompagnant la notion de vache en Inde sera différent du vocabulaire utilisé en Espagne, lui-même différent de celui utilisé en Argentine...).
Pour cela, nous introduisons la notion de similarité intra-classe.

Définition 11 (Similarité intra-classe)

Similarité intra-classe : similarité entre documents d'une même classe.

À noter qu'une classe très générale ("chat", "chien", "canard"), n'implique pas nécessairement des documents différents, de la même façon qu'une classe que l'on pourrait qualifier de très spécifique n'implique pas obligatoirement une similarité entre documents la composant (par exemple "l'alimentation de la vache", "la perception de la vache à travers le monde") cependant il est très probable que dans le cas de classe très spécifique, le vocabulaire soit naturellement limité.

Ainsi, pour résumer, nous pouvons décrire un corpus comme un ensemble de classes plus ou moins similaire (similarité inter-classes), chaque classe étant composée d'un ensemble de documents plus ou moins similaires (similarité intra-classe). Nous pouvons illustrer cette définition en projetant ces similarités dans l'espace en positionnant les centres de gravité des classes en fonction de la similarité entre les classes et la taille de la classe variant en fonction de la similarité entre documents de la classe. Par exemple, dans la figure 9.2, les cercles symbolisent les classes, les points symbolisent les documents et la similarité inter-classes est symbolisée par une flèche alors que les similarités intra-classes sont représentées par l'écart entre les points d'une même classe.

La première question sous-jacente à cette représentation d'un corpus est comment mesurer la similarité entre deux classes ou deux documents ?

9.2 Le choix de la mesure de similarité

Généralement, la similarité est évaluée soit par une mesure de similarité, soit par une distance. Les similarités peuvent être exprimées sous forme de distances ($D = 1 - S$) tout comme les distances peuvent être exprimées en similarités

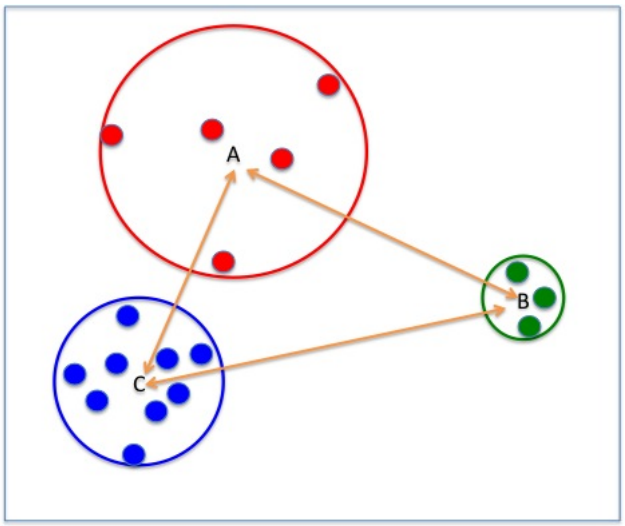


Figure 9.2: Similarité projetée dans l'espace

(dans ce cas, il faut normaliser entre 0 et 1 les valeurs qui peuvent varier entre 0 et l'infini).

De nombreuses mesures de similarité ou de distances existent dans la littérature et sont, par exemple, comparées dans [175, 56, 93, 24]. Certaines mesures de similarité ou de distances sont applicables uniquement à des attributs binaires (Jaccard, Dice) là où d'autres sont applicables aussi sur des attributs non binaires (Distance Euclidienne, Cosinus, Tanimoto).

Nous nous focalisons dans la suite sur le coefficient de Tanimoto qui présente l'avantage d'être borné entre 0 et 1 et qui permet de prendre en considération la fréquence des descripteurs et non uniquement leur absence ou présence. Nous précisons néanmoins que toutes autres mesures de similarités applicables sur des attributs non binaires pourraient convenir (par exemple la similarité Cosinus).

Le coefficient de Tanimoto [132], aussi appelé coefficient étendu de Jaccard, est la forme généralisée du coefficient de Jaccard [59]. Contrairement au coefficient de Jaccard, il est adapté aux attributs non binaires. Il mesure la similarité entre deux ensembles en comparant la taille de l'intersection par rapport à la taille de l'union des deux ensembles. Le coefficient de Tanimoto vaut 1 lorsque les ensembles sont identiques, 0 lorsqu'ils sont disjoints.

Pour un document, le coefficient de Tanimoto compare ainsi le poids de la somme des descripteurs communs à la somme des poids des descripteurs qui sont présents dans l'un des deux documents, mais qui ne sont pas des descripteurs communs. La définition formelle est :

Classe	Documents	Descripteurs
Bovidae	bov_1	animal-compagnie-boeuf-génisse-vachette-yack-yack-yack
	bov_2	animal-boeuf-taureau
	bov_3	animal-yack
	bov_4	animal-herbe-marguerite-ruminer-train-vachette
	bov_5	animal-corne-mamifère-peau-yack
Canidae	can_1	animal-compagnie-chien-chienne-chiot-toutou
	can_2	animal-compagnie-chiot-chiot-toutou-toutou
	can_3	animal-compagnie-milou-medor-rantanplan-pif
	can_4	animal-carnivore-chacal-mammifère-renard
	can_5	balle-chien-croquette-laisse
	can_6	cage-carnivore-griffe-loup-mammifère
	can_7	animal-chien-compagnie-spa
Felidae	fel_1	animal-chat-compagnie-matou-matou-minet
	fel_2	animal-compagnie-matou-minou-minou-minou
	fel_3	animal-compagnie-matou-minou-minou-minou
	fel_4	animal-carnivore-lynx-mammifère-puma
	fel_5	balle-croquette-laisse-minou
	fel_6	cage-carnivore-griffe-shereKhan-tigre
	fel_7	animal-chat-compagnie-mammifère-spa
Ornithorhynchidae	ort_1	animal-australie-carnivore-mammifère-ornithorynque-toto
	ort_2	australie-hexley-mammifère-ornithorynque
	ort_3	animal-australie-carnivore-mammifère-ornithorynque-poil-venin
	ort_4	animal-australie-bec-carnivore-eau-mammifère-œuf-ornithorynque-palme
Anatidae	ana_1	cygne-cygne-eau-mare-vilain
	ana_2	animal-canard-confit-foie-magret
	ana_3	aile-animal-bec-œuf-oie-palme-plume
	ana_4	canardo-daffy-donald-saturnin
	ana_5	animal-canardeau-cancane-cane-caneton-canette

Table 9.1: Corpus exemple

Définition 12 (Coefficient de Tanimoto)

$$S_J(d_1, d_2) = \frac{\sum_{k=1}^{N}(w_{k1} \times w_{k2})}{\sum_{k=1}^{N} w_{k1}^2 + \sum_{k=1}^{N} w_{k2}^2 - \sum_{k=1}^{N}(w_{k1} \times w_{k2})}$$

où w_{k1} correspond au poids du $k^{ème}$ mot dans le document d_1 et w_{k2} correspond au poids du $k^{ème}$ mot dans le document d_2

Si les ensembles sont composés d'attributs binaires, le coefficient de Tanimoto se réduit au coefficient de Jaccard.

9.3 Mesurer les similarités inter-classes et intra-classes

Reprenons et complétons le corpus exemple utilisé pour illustrer la première partie de ce manuscrit en y ajoutant des documents aux classes existantes ainsi que deux classes supplémentaires.
En se basant sur la classification scientifique des espèces, nous pouvons supposer que le corpus exemple est maintenant composé :

- de classes plus similaires que d'autres (les familles Felidae et Canidae appartiennent toutes deux à l'ordre Carnivora) ;
- de classes au vocabulaire plus large que d'autres (l'ornithorynque est la seule espèce présente dans la famille des Ornithorhynchidae alors que la famille des Bovidae, par exemple, est composée de plus de 50 espèces, réparties dans 9 sous-familles et 42 genres).

9.3.1 Similarités inter-classes

Nous calculons les similarités S_J pour chaque paire de classes ($S_J(Canidae, Felidae)$, $S_J(Canidae, Bovidae)$...). Par exemple, la représentation *sac de mots* des classes Canidae et Felidae est :

- $V_{Canidae}$:{animal (5), balle (1), cage (1), carnivore (2), chacal (1), chien (3), chienne (1), chiot (3), compagnie (4), croquette (1), griffe (1), laisse (1), loup (1), mammifère (2), medor (1), milou (1), pif (1), rantanplan (1), renard (1), spa (1), toutou (3)}
- $V_{Felidae}$:{animal (5), balle (1), cage (1), carnivore (2), chat (2), compagnie (4), croquette (1), griffe (1), laisse (1), lynx (1), mammifère (2), matou (4), minet (1), minou (7), puma (1), shereKhan (1), spa (1), tigre (1)}

Le détail des calculs pour $S_J(Canidae, Felidae)$ est donné dans le tableau 9.2.
Et la similarité entre les classes $Canidae$ et $Felidae$ est obtenue en calculant le coefficient de Tanimoto avec :

Exemple 9 ($S_J(V_{Canidae}, V_{Felidae})$)

$$S_J(V_{Canidae}, V_{Felidae}) = \frac{\sum_{k=1}^{N}(w_{k1} \times w_{k2})}{\sum_{k=1}^{N} w_{k1}^2 + \sum_{k=1}^{N} w_{k2}^2 - \sum_{k=1}^{N}(w_{k1} \times w_{k2})} = \frac{55}{90+129-55} = 0.34$$

Nous résumons les différentes similarités entre les classes observées dans notre exemple (Tableau 9.3).
Ainsi, le nombre de similarités inter-classes à calculer, que nous nommons $nsim_{Inter}$ (Définition n°13), va dépendre du nombre de classes du corpus. Le nombre de similarités inter-classes à calculer est égal au nombre d'arêtes du graphe complet représentant les $|C|$ classes du corpus.

Définition 13 ($nsim_{Inter}$)

$nsim_{Inter}$: nombre de similarités inter-classes à calculer pour un corpus donné.

Avec : $nsim_{Inter} = \frac{|C|(|C|-1)}{2}$

	Canidae	Felidae	$w_{k1} \times w_{k2}$	w_{k1}^2	w_{k2}^2
animal	5	5	25	25	25
balle	1	1	1	1	1
cage	1	1	1	1	1
carnivore	2	2	4	4	4
chacal	1	0	0	1	0
chat	0	2	0	0	4
chien	3	0	0	9	0
chienne	1	0	0	1	0
chiot	3	0	0	9	0
compagnie	4	4	16	16	16
croquette	1	1	1	1	1
griffe	1	1	1	1	1
laisse	1	1	1	1	1
loup	1	0	0	1	0
lynx	0	1	0	0	1
mammifère	2	2	4	4	4
matou	0	4	0	0	16
medor	1	0	0	1	0
milou	1	0	0	1	0
minet	0	1	0	0	1
minou	0	7	0	0	49
pif	1	0	0	1	0
puma	0	1	0	0	1
rantanplan	1	0	0	1	0
renard	1	0	0	1	0
shereKhan	0	1	0	0	1
spa	1	1	1	1	1
tigre	0	1	0	0	1
toutou	3	0	0	9	0
$\sum_{k=1}^{N}$			55	90	129

Table 9.2: Représentation *sac de mots* des classes Felidae et Canidae

	Anatidae	Bovidae	Canidae	Felidae	Ornithorhynchidae
Anatidae		0.17	0.14	0.10	0.14
Bovidae	0.17		0.22	0.17	0.12
Canidae	0.14	0.22		0.34	0.21
Felidae	0.10	0.17	0.34		0.17
Ornithorhynchidae	0.14	0.12	0.21	0.17	

Table 9.3: Calcul des similarités inter-classes

$S_J(x,y)$	fel_1	fel_2	$w_{k1} \times w_{k2}$	w_{k1}^2	w_{k2}^2
animal	1	1	1	1	1
chat	1	0	0	1	0
compagnie	1	1	1	1	1
matou	2	1	2	4	1
minet	1	0	0	1	0
minou	0	3	0	0	9
$\sum_{k=1}^{N}$			4	8	12

Table 9.4: Représentation *sac de mots*s des documents fel_1 et fel_2

Dans notre corpus exemple composé de 5 classes, le nombre de similarités inter-classes calculé est :

Exemple 10 ($nsim_{Inter}$)

$nsim_{Inter} = \frac{5(5-1)}{2} = 10$

Cette représentation est utile pour étudier les similarités entre les classes du corpus mais elle ne permet pas de prendre en compte les similarités observées au sein même des classes. C'est pourquoi nous calculons les similarités intra-classes dans la section suivante.

9.3.2 Similarités intra-classes

En introduction de ce chapitre, nous avons défini la similarité intra-classe comme étant la similarité entre documents d'une même classe. Par exemple, si nous considérons la classe Felidae, composée des documents suivants :

- fel_1 {animal-chat-compagnie-matou-matou-minet}
- fel_2 {animal-compagnie-matou-minou-minou-minou}
- fel_3 {animal-compagnie-matou-minou-minou-minou}
- fel_4 {animal-carnivore-lynx-mammifère-puma}
- fel_5 {balle-croquette-laisse-minou}
- fel_6 {cage-carnivore-griffe-shereKhan-tigre}
- fel_7 {animal-chat-compagnie-mammifère-spa}

Cette classe est composée de sept documents et nous pouvons constater que les documents fel_2 : {animal-compagnie-matou-minou-minou-minou} et fel_3 : {animal-compagnie-matou-minou-minou-minou} sont identiques. À l'opposé, les documents fel_2 : {animal-compagnie-matou-minou-minou-minou} et fel_6 : {cage-carnivore-griffe-shereKhan-tigre} ne partagent aucun descripteur en commun. Nous calculons les similarités S_J pour chaque paire de documents de la classe ($S_J(fel\_1, fel\_2)$, $S_J(fel\_1, fel\_3)$...). Par exemple pour $S_J(fel\_1, fel\_2)$ le détail des calculs est donné dans le tableau 9.4.
Et la similarité entre les documents fel_1 et fel_2 est obtenue en calculant le coefficient de Tanimoto avec :

$S_J(x,y)$	fel_1	fel_2	fel_3	fel_4	fel_5	fel_6	fel_7
fel_1		0.25	0.25	0.08	0	0	0.3
fel_2	0.25		1	0.063	0.23	0	0.13
fel_3	0.25	1		0.063	0.23	0	0.13
fel_4	0.083	0.063	0.063		0	0.11	0.25
fel_5	0	0.23	0.23	0		0	0
fel_6	0	0	0	0.11	0		0
fel_7	0.3	0.13	0.13	0.25	0	0	

Table 9.5: Calcul des similarités intra-classes pour la classe Felidae

Exemple 11 ($S_J(V_{fel\_1}, V_{fel\_2})$)

$$S_J(V_{fel\_1}, V_{fel\_2}) = \frac{\sum_{k=1}^{N}(w_{k1} \times w_{k2})}{\sum_{k=1}^{N} w_{k1}^2 + \sum_{k=1}^{N} w_{k2}^2 - \sum_{k=1}^{N}(w_{k1} \times w_{k2})} = \frac{4}{8+12-4} = 0.25$$

Nous résumons les différentes similarités observées dans notre étude de cas dans le tableau 9.5.
Comme attendu, nous pouvons remarquer que $S_J(V_{fel\_2}, V_{fel\_3}) = 1$ (documents identiques) et $S_J(V_{fel\_2}, V_{fel\_6}) = 0$ (documents ne partageant aucun descripteur).
Pour une classe n donnée, le nombre de similarités intra-classes à calculer, que nous nommons $nsim^n_{Intra}$ (Définition n°15), va dépendre du nombre de documents la composant. Il est égal au nombre d'arêtes du graphe complet représentant les $|d|$ documents de la classe.

Définition 14 ($nsim^n_{Intra}$)

$nsim^n_{Intra}$: nombre de similarités intra-classes à calculer pour une classe n donnée.

Avec : $nsim^n_{Intra} = \frac{|d|(|d|-1)}{2}$

Dans notre corpus exemple, pour la classe Felidae composée de 7 documents, le nombre de similarités intra-classes calculé est :

Exemple 12 ($nsim^{Felidae}_{Intra}$)

$$nsim^{Felidae}_{Intra} = \frac{7(7-1)}{2} = 21$$

Le nombre de similarités intra-classes $nsim^n_{Intra}$ peut varier d'une classe à l'autre et nous synthétisons dans le tableau 9.6 les 5 $nsim^n_{Intra}$ calculés pour les 5 classes de notre exemple.

Classe	Nombre de documents	$nsim^n_{Intra}$
Bovidae	5	10
Canidae	7	21
Felidae	7	21
Ornithorhynchidae	4	6
Anatidae	5	10

Table 9.6: Nombre de similarités intra-classes

Ainsi pour un corpus composé de $|C|$ classes, le nombre de similarités intra-classes total à calculer, que nous nommons $nsim_{Intra}$ (Définition n°15), correspond à la somme des $|C|$ $nsim^n_{Intra}$.

Définition 15 ($nsim_{Intra}$)

$nsim_{Intra}$: nombre total de similarités intra-classe à calculer.

Avec : $nsim_{Intra} = \sum_{n=1}^{|C|} nsim^n_{Intra}$.

Dans notre exemple, il y a donc 68 similarités intra-classes à calculer.

Exemple 13 ($nsim_{Intra}$)

$nsim_{Intra} = \sum_{n=1}^{|C|} nsim^n_{Intra}$
$= nsim^{Bovidae}_{Intra} + nsim^{Canidae}_{Intra} + nsim^{Felidae}_{Intra} + nsim^{Ornithorhynchidae}_{Intra} + nsim^{Anatidae}_{Intra}$
$= 10 + 21 + 21 + 4 + 10 = 68$

Nous donnons les 68 similarités intra-classes dans le tableau 9.7.

Anatidae		Bovidae		Canidae		Felidae		Ornithorhynchidae	
$S_J(ana\_1, ana\_2)$	0	$S_J(bov\_1, bov\_2)$	0.13	$S_J(can\_1, can\_2)$	0.60	$S_J(fel\_1, fel\_2)$	0.25	$S_J(ort\_1, ort\_2)$	0.43
$S_J(ana\_1, ana\_3)$	0	$S_J(bov\_1, bov\_3)$	0.33	$S_J(can\_1, can\_3)$	0.20	$S_J(fel\_1, fel\_3)$	0.25	$S_J(ort\_1, ort\_3)$	0.63
$S_J(ana\_1, ana\_4)$	0	$S_J(bov\_1, bov\_4)$	0.11	$S_J(can\_1, can\_4)$	0.10	$S_J(fel\_1, fel\_4)$	0.08	$S_J(ort\_1, ort\_4)$	0.50
$S_J(ana\_1, ana\_5)$	0	$S_J(bov\_1, bov\_5)$	0.27	$S_J(can\_1, can\_5)$	0.11	$S_J(fel\_1, fel\_5)$	0	$S_J(ort\_2, ort\_3)$	0.38
$S_J(ana\_2, ana\_3)$	0.09	$S_J(bov\_2, bov\_3)$	0.25	$S_J(can\_1, can\_6)$	0	$S_J(fel\_1, fel\_6)$	0	$S_J(ort\_2, ort\_4)$	0.30
$S_J(ana\_2, ana\_4)$	0	$S_J(bov\_2, bov\_4)$	0.13	$S_J(can\_1, can\_7)$	0.43	$S_J(fel\_1, fel\_7)$	0.30	$S_J(ort\_3, ort\_4)$	0.45
$S_J(ana\_2, ana\_5)$	0.10	$S_J(bov\_2, bov\_5)$	0.14	$S_J(can\_2, can\_3)$	0.14	$S_J(fel\_2, fel\_3)$	1		
$S_J(ana\_3, ana\_4)$	0	$S_J(bov\_3, bov\_4)$	0.14	$S_J(can\_2, can\_4)$	0.07	$S_J(fel\_2, fel\_4)$	0.06		
$S_J(ana\_3, ana\_5)$	0.08	$S_J(bov\_3, bov\_5)$	0.40	$S_J(can\_2, can\_5)$	0	$S_J(fel\_2, fel\_5)$	0.23		
$S_J(ana\_4, ana\_5)$	0	$S_J(bov\_4, bov\_5)$	0.10	$S_J(can\_2, can\_6)$	0	$S_J(fel\_2, fel\_6)$	0		
				$S_J(can\_2, can\_7)$	0.17	$S_J(fel\_2, fel\_7)$	0.13		
				$S_J(can\_3, can\_4)$	0.10	$S_J(fel\_3, fel\_4)$	0.06		
				$S_J(can\_3, can\_5)$	0	$S_J(fel\_3, fel\_5)$	0.23		
				$S_J(can\_3, can\_6)$	0	$S_J(fel\_3, fel\_6)$	0		
				$S_J(can\_3, can\_7)$	0.25	$S_J(fel\_3, fel\_7)$	0.13		
				$S_J(can\_4, can\_5)$	0	$S_J(fel\_4, fel\_5)$	0		
				$S_J(can\_4, can\_6)$	0.25	$S_J(fel\_4, fel\_6)$	0.11		
				$S_J(can\_4, can\_7)$	0.13	$S_J(fel\_4, fel\_7)$	0.25		
				$S_J(can\_5, can\_6)$	0	$S_J(fel\_5, fel\_6)$	0		
				$S_J(can\_5, can\_7)$	0.14	$S_J(fel\_5, fel\_7)$	0		
				$S_J(can\_6, can\_7)$	0	$S_J(fel\_6, fel\_7)$	0		

Table 9.7: Les 68 similarités intra-classes issues de notre exemple

Le nombre total $nsim$ de similarités inter-classes et intra-classes calculées pour un corpus composé de |C| classes est donné dans :

Définition 16 ($nsim$)

$nsim$: nombre total de similarités inter-classes et intra-classes.

Avec : $nsim = nsim_{Inter} + nsim_{Intra} = \frac{|C|(|C|-1)}{2} + \sum_{n=1}^{|C|} \frac{|d_n|(|d_n|-1)}{2}$.

Ainsi pour notre corpus exemple, $nsim = 78$ et correspond aux 10 similarités inter-classes ajoutées aux 68 similarités intra-classes.

Exemple 14 ($nsim$)

$nsim = nsim_{Inter} + nsim_{Intra} = 10 + 68 = 78$

Cette représentation implique qu'en cas d'ajout ou de suppression d'un document ou d'une classe, le nombre de similarités calculé $nsim$ sera différent. Or l'objectif est de pouvoir comparer deux corpus en utilisant leurs similarités comme méta-descripteurs et pour cela il est nécessaire d'utiliser un nombre identique de méta-descripteurs. Nous détaillons dans la section suivante comment agréger les $nsim$ similarités en un nombre fini de méta-descripteurs indépendamment de la taille du corpus.

9.4 D'un nombre variable de similarités à un nombre fini de méta-descripteurs

Dans la section précédente, il est établi qu'un corpus est défini par $nsim$ similarités, réparties entre $nsim_{Inter}$ similarités inter-classes (10 dans notre exemple) et $nsim_{Intra}$ similarités intra-classes (68 dans notre exemple). Tout corpus peut donc être décrit par deux ensembles de similarités indépendants, un ensemble de similarités inter-classes et un ensemble de similarités intra-classes.

Définition 17 (S_{Inter} et S_{Intra})

S_{Inter} : ensemble des similarités inter-classes d'un corpus

S_{Intra} : ensemble des similarités intra-classes d'un corpus

Nous pouvons maintenant nous interroger sur la manière de comparer n ensembles S_{Inter} calculés à partir de n corpus différents sachant que les ensembles peuvent être de tailles $|S_{Inter}|$ différentes. Nous pensons que cette opération est possible en transformant les n ensembles de taille variable en n ensembles de taille finie et identique.
En effet, un ensemble composé d'un nombre variable de valeurs peut être décrit au moyen d'un nombre fini de statistiques descriptives simples tels que la moyenne, la médiane, le mode, le maximum, le minimum, l'écart type, la variance... Dans ce cadre, nous introduisons quelques définitions :

Définition 18 ($S_n - Stat$)

Soit D, un ensemble de statistiques descriptives.
$S_n - Stat$ correspond à l'ensemble des valeurs obtenues sur les éléments composant S_n par application des statistiques descriptives définies dans D.
Le nombre de valeurs $|S_n - Stat|$ correspond au nombre de statistiques descriptives $|D|$ retenues.

Considérons par exemple l'ensemble D composé de 4 statistiques descriptives : {moyenne, variance, minimum, maximum}. Soit S_1 : (0.1, 0.4, 0.5, 0.7, 0.9), un ensemble de 5 valeurs numériques et S_2 : (0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.3, 0.4, 0.4, 0.5, 0.5, 0.6, 0.6, 0.7, 0.7, 0.8, 0.8, 0.9, 0.9, 1), un ensemble de 20 valeurs numériques comprises entre 0 et 1. Par calcul des $S_n - Stat$, il devient alors possible de comparer deux ensembles S_1 et S_2 de taille différente en comparant $S_1 - Stat$ et $S_2 - Stat$ comme illustré dans l'Exemple 15.

Exemple 15 ($S_1 - Stat$ et $S_2 - Stat$)

$S_1 - Stat$: (moyenne(S_1), variance(S_1), minimum(S_1), maximum(S_1))
$S_1 - Stat$: (0.5, 0.08, 0.1, 0.9)
$|S_1 - Stat| = 4$

$S_2 - Stat$: (moyenne(S_2), variance(S_2), minimum(S_2), maximum(S_2))
$S_2 - Stat$: (0.5, 0.085, 0, 1)
$|S_2 - Stat| = 4$

Il est alors possible de comparer $S_1 - Stat$ et $S_2 - Stat$ car la taille des nouveaux ensembles est identique ($|S_1 - Stat|$=$|S_2 - Stat|$=4) bien que les deux ensembles initiaux soient de taille différente ($|S_1| = 5$ et $|S_2| = 20$).
La comparaison n'aura de sens que si les ensembles S_1 et S_2 sont composés de valeurs avec un ordre de grandeur cohérent (par exemple borné entre 0 et 1). En

effet, l'ordre de grandeur de $S_n - Stat$ est corrélé aux valeurs de S_n et si S_1 et S_2 sont composés de valeurs avec un ordre de grandeur cohérent alors $S_1 - Stat$ et $S_2 - Stat$ seront composés de valeurs avec un ordre de grandeur cohérent (et donc comparable), même si le nombre de valeurs $|S_1 - Stat|$ est très largement supérieur à $|S_2 - Stat|$.

Dans la section suivante, nous étudions l'application de ces définitions sur notre cas d'étude.

9.4.1 Application aux similarités inter-classes

Nous nous intéressons pour commencer à l'agrégation des similarités inter-classes. En utilisant D : {moyenne, variance, minimum, maximum}, nous pouvons définir $S_{Inter} - Stat$ à partir de S_{Inter} de taille $nsim_{Inter}$ défini dans le tableau 9.3.

Exemple 16 (S_{Inter} et $S_{Inter} - Stat$)

S_{Inter} : $(0.17, 0.14, 0.10, 0.14, 0.22, 0.17, 0.12, 0.34, 0.21, 0.17)$
$|S_{Inter}| = nsim_{Inter} = 10$

$S_{Inter} - Stat$:(moyenne(S_{Inter}), variance(S_{Inter}), minimum(S_{Inter}), maximum(S_{Inter}))
avec :
$moyenne(S_{Inter})$=moyenne(0.17, 0.14, 0.10, 0.14, 0.22, 0.17, 0.12, 0.34, 0.21, 0.17)=0.18
$variance(S_{Inter})$=variance(0.17, 0.14, 0.10, 0.14, 0.22, 0.17, 0.12, 0.34, 0.21, 0.17)=0.005
$minimum(S_{Inter})$=minimum(0.17, 0.14, 0.10, 0.14, 0.22, 0.17, 0.12, 0.34, 0.21, 0.17)=0.10
$maximum(S_{Inter})$=maximum(0.17, 0.14, 0.10, 0.14, 0.22, 0.17, 0.12, 0.34, 0.21, 0.17)=0.34

$S_{Inter} - Stat$:(0.18, 0.005, 0.10, 0.34)
$|S_{Inter} - Stat| = 4$

Ces valeurs représentent la distance moyenne entre les classes du corpus, la variance des distances entre les classes ou encore la distance minimum et maximum entre les classes du corpus et permettent de savoir :

- si les classes sont en moyenne similaires les unes des autres (moyenne des distances inter-classes),
- si les distances entre classes sont similaires (variance des distances inter-classes),
- quel est l'écart minimum entre classes (minimum des distances inter-classes),
- quel est l'écart maximum entre classes (maximum des distances inter-classes).

Le nombre d'éléments $|S_{Inter} - Stat|$ est indépendant du nombre d'éléments de l'ensemble S_{Inter}, qui lui, nous le rappelons dépend du nombre de classes (Section 9.3.1). Ainsi, indépendamment du nombre de classes du corpus, il est possible de définir la représentation inter-classe par un nombre fini D de valeurs, où D correspond au nombre de descripteurs statistiques choisis pour décrire le corpus.

Nous avons rappelé que tout corpus peut être décrit par deux ensembles de similarités indépendants, un ensemble de similarités inter-classes S_{Inter} et un ensemble de similarités intra-classes S_{Intra}. Nous venons de décrire comment agréger les $nsim_{Inter}$ similarités inter-classes S_{Inter} en un nombre fini de valeurs indépendant du corpus, regardons maintenant comment agréger les $nsim_{Intra}$ similarités intra-classes S_{Intra}.

9.4.2 Application aux similarités intra-classes

Nous rappelons que dans notre exemple, nous avons calculé 68 similarités intra-classes. Tout d'abord, à partir du tableau 9.7 nous pouvons définir S_{Intra} et $|S_{Intra}|$.

Exemple 17 (S_{Intra} et $|S_{Intra}|$)

S_{Intra} : (0.13, 0.33, 0.11, 0.27, 0.25, 0.13, 0.14, 0.14, 0.40, 0.10, 0.60, 0.20, 0.10, 0.11, 0, 0.43, 0.14, 0.07, 0, 0, 0.17, 0.10, 0, 0, 0.25, 0, 0.25, 0.13, 0, 0.14, 0, 0.25, 0.25, 0.08, 0, 0, 0.30, 1, 0.06, 0.23, 0, 0.13, 0.06, 0.23, 0, 0.13, 0, 0.11, 0.25, 0, 0, 0, 0.43, 0.63, 0.50, 0.38, 0.30, 0.45, 0, 0, 0, 0, 0.09, 0, 0.10, 0, 0.08, 0)

$|S_{Intra}| = nsim_{Intra} = 68$

Nous pourrions envisager d'agréger les $nsim_{Intra}$ similarités intra-classes comme nous l'avons fait pour les $nsim_{Inter}$ similarités inter-classes, mais nous pensons qu'il est possible d'obtenir une vision plus précise de la réalité. En effet, nous avons établi que les $nsim_{Intra}$ similarités intra-classes du corpus étaient en fait la somme des $nsim^n_{Intra}$ similarités intra-classes indépendantes entre elles. Ainsi, S_{Intra} peut être décomposé en $|C|$ sous-ensembles indépendants S^n_{Intra} de taille $nsim^n_{Intra}$ où $n \in (1, ..., |C|)$.

Exemple 18 (S^n_{Intra} et $|S^n_{Intra}|$)

S_{Intra} : $S^{Bovidae}_{Intra} \cup S^{Canidae}_{Intra} \cup S^{Felidae}_{Intra} \cup S^{Ornithorhynchidae}_{Intra} \cup S^{Anatidae}_{Intra}$

$S^{Bovidae}_{Intra}$: (0.13, 0.33, 0.11, 0.27, 0.25, 0.13, 0.14, 0.14, 0.40, 0.10)

$|S^{Bovidae}_{Intra}| = nsim^{Bovidae}_{Intra}$=10

$S^{Canidae}_{Intra}$: (0.60, 0.20, 0.10, 0.11, 0, 0.43, 0.14, 0.07, 0, 0, 0.17, 0.10, 0, 0, 0.25, 0, 0.25, 0.13, 0, 0.14, 0)

$|S^{Canidae}_{Intra}| = nsim^{Canidae}_{Intra}$=21

$S^{Felidae}_{Intra}$: (0.25, 0.25, 0.08, 0, 0, 0.30, 1, 0.06, 0.23, 0, 0.13, 0.06, 0.23, 0, 0.13, 0, 0.11, 0.25, 0, 0, 0)

$|S^{Felidae}_{Intra}| = nsim^{Felidae}_{Intra}$=21

$S^{Ornithorhynchidae}_{Intra}$: (0.43, 0.63, 0.50, 0.38, 0.30, 0.45)

$|S^{Ornithorhynchidae}_{Intra}| = nsim^{Ornithorhynchidae}_{Intra}$=6

$S^{Anatidae}_{Intra}$: (0, 0, 0, 0, 0.09, 0, 0.10, 0, 0.08, 0)

$|S^{Anatidae}_{Intra}| = nsim^{Anatidae}_{Intra}$=10

Considérer les $nsim_{Intra}$ similarités intra-classes comme un seul et même ensemble revient à faire abstraction du comportement spécifique de chaque classe et revient à considérer que toutes les classes d'un même corpus sont composées de documents à la similarité identique. Or certaines classes sont composées de documents différents, d'autres de documents similaires et d'autres sont un mélange de documents similaires et différents (cf. Section 9.1). Contrairement à la similarité inter-classes où l'objectif consistait à agréger un seul ensemble composé d'un nombre variable de valeurs, l'objectif ici est d'agréger $|C|$ ensembles composés d'un nombre variable de valeurs. Nous souhaitons alors obtenir un nombre fini d'ensembles composés d'un nombre fini de valeurs. Pour cela, nous proposons une approche en deux étapes. Sur notre exemple, en conservant D : {moyenne, variance, minimum, maximum}, nous pouvons tout d'abord calculer 5 $S^n_{Inter} - Stat$ indépendamment pour chacune des classes à partir des 5 S^n_{Inter} telles que définies dans le tableau 9.7.

Exemple 19 ($S^n_{Intra} - Stat$)

$S^n_{Intra} - Stat$:(moyenne(S^n_{Intra}), variance(S^n_{Intra}), minimum(S^n_{Intra}), maximum(S^n_{Intra}))

$S^{Bovidae}_{Intra}$: (0.13, 0.33, 0.11, 0.27, 0.25, 0.13, 0.14, 0.14, 0.40, 0.10)
$S^{Bovidae}_{Intra} - Stat$:(0.20, 0.001, 0.10, 0.40)

$S^{Canidae}_{Intra}$: (0.60, 0.20, 0.10, 0.11, 0, 0.43, 0.14, 0.07, 0, 0, 0.17, 0.10, 0, 0, 0.25, 0, 0.25, 0.13, 0, 0.14, 0)
$S^{Canidae}_{Intra} - Stat$:(0.13, 0.024, 0, 0.60)

$S^{Felidae}_{Intra}$: (0.25, 0.25, 0.08, 0, 0, 0.30, 1, 0.06, 0.23, 0, 0.13, 0.06, 0.23, 0, 0.13, 0, 0.11, 0.25, 0, 0, 0)
$S^{Felidae}_{Intra} - Stat$:(0.15, 0.05, 0, 1)

$S^{Ornithorhynchidae}_{Intra}$: (0.43, 0.63, 0.50, 0.38, 0.30, 0.45)
$S^{Ornithorhynchidae}_{Intra} - Stat$:(0.45, 0.0125, 0.30, 0.63)

$S^{Anatidae}_{Intra}$: (0, 0, 0, 0, 0.09, 0, 0.10, 0, 0.08, 0)
$S^{Anatidae}_{Intra} - Stat$:(0.03, 0.002, 0, 0.10)

Nous constatons que la taille des 5 ensembles $S^n_{Intra} - Stat$ est identique (et égale au nombre de descripteurs statistiques de D, i.e. 4). Nous avons mesuré la similarité moyenne au sein de chaque classe, la variance au sein de chaque classe ou encore le minimum et le maximum des similarités entre documents au sein de chaque classe. Cependant, si nous avons résumé un nombre variable de valeurs (propre à chaque classe) en un nombre fini de valeurs (4) pour chacune des classes, le nombre total de valeurs calculé reste dépendant du nombre de classes (5×4 dans notre exemple). Or nous souhaitons nous prémunir de la variation du nombre de classes en fonction des corpus.

Chacun des $S^n_{Intra} - Stat$ étant composé d'un même nombre de valeurs et chacune de ces valeurs étant issue d'un même traitement statistique (moyenne, variance, minimum et maximum dans notre exemple), nous proposons d'agréger l'ensemble des moyennes des $S^n_{Intra} - Stat$ puis l'ensemble des variances, et des minimums et pour finir des maximums. Ainsi au lieu de considérer des ensembles $S^n_{Intra} - Stat$ où n correspond à une classe, nous considérons S^D_{Intra} où D correspond à un descripteur statistique. Par exemple, l'ensemble $S^{moyenne}_{Intra}$ contient les valeurs moyenne($S^{Bovidae}_{Intra}$), moyenne($S^{Canidae}_{Intra}$), moyenne($S^{Felidae}_{Intra}$), moyenne($S^{Ornithorhynchidae}_{Intra}$), moyenne($S^{Anatidae}_{Intra}$)). Ainsi à partir des $|C|$ ensembles $S^n_{Intra} - Stat$ de $|D|$ valeurs, il est possible de définir $|D|$ ensembles S^D_{Intra} de $|C|$ valeurs puis de calculer $S^D_{Intra} - Stat$ à partir des ensembles S^D_{Intra} comme nous l'avions fait pour calculer

$S^n_{Intra} - Stat$ à partir des ensembles S^n_{Intra}.
Pour notre étude de cas, nous calculons $S^{moyenne}_{Intra}$, $S^{variance}_{Intra}$, $S^{minimum}_{Intra}$ et $S^{maximum}_{Intra}$.

Exemple 20 ($S^D_{Intra} - Stat$)

$S^{moyenne}_{Intra}$: (0.20, 0.13, 0.15, 0.45, 0.03)
$S^{moyenne}_{Intra} - Stat$: (0.19, 0.02, 0.03, 0.45)

$S^{variance}_{Intra}$: (0.01, 0.02, 0.05, 0.01, 0)
$S^{variance}_{Intra} - Stat$: (0.02, 0, 0, 0.05)

$S^{minimum}_{Intra}$: (0.10, 0, 0, 0.30, 0)
$S^{minimum}_{Intra} - Stat$: (0.08, 0.02, 0, 0.30)

$S^{maximum}_{Intra}$: (0.40, 0.60, 1, 0.63, 0.10)
$S^{maximum}_{Intra} - Stat$: (0.55, 0.11, 0.10, 1)

- $S^{moyenne}_{Intra}$ représente l'ensemble des distances moyennes observées au sein de chaque classe du corpus.
- $S^{moyenne}_{Intra} - Stat$ représente la moyenne des distances moyennes au sein des classes, la variance des distances moyennes au sein des classes ou encore le minimum et le maximum des distances moyennes au sein des classes.
- $S^{variance}_{Intra}$ représente l'ensemble des variances observées au sein de chaque classe du corpus.
- $S^{variance}_{Intra} - Stat$ représente aussi la moyenne des variances au sein des classes, la variance des variances au sein des classes ou encore le minimum et le maximum des variances au sein des classes.
- $S^{minimum}_{Intra}$ et $S^{maximum}_{Intra}$ représentent l'ensemble des minimums et des maximums observés au sein de chaque classe du corpus.
- $S^{minimum}_{Intra} - Stat$ et $S^{maximum}_{Intra} - Stat$ représentent les moyennes des distances minimums et maximums au sein des classes, la variance des distances minimums et maximums au sein des classes ou encore le minimum des distances minimums et maximums au sein des classes et le maximum des distances minimums et maximums au sein des classes.

Dans notre exemple, ils permettent de savoir pour le corpus :
- La distance moyenne la plus petite (le minimum des moyennes) ;
- La distance moyenne la plus grande (le maximum des moyennes) ;
- La moyenne et la variance des distances moyennes (la moyenne et la variance des moyennes) ;
- La variance des distances moyenne la plus petite (le minimum des variances) ;
- La variance des distances moyenne la plus grande (le maximum des variances) ;

- La moyenne et la variance des variances des distances moyennes (la moyenne et la variance des variances) ;
- La distance minimum la plus petite (le minimum des minimums) ;
- La distance minimum la plus grande (le maximum des minimums) ;
- La moyenne et la variance des distances minimum (la moyenne et la variance des minimums) ;
- La distance maximum la plus petite (le minimum des maximums) ;
- La distance maximum la plus grande (le maximum des maximums) ;
- La moyenne et la variance des distances maximum (la moyenne et la variance des maximums).

Pour résumer, nous proposons d'agréger les similarités intra-classes en deux étapes successives comme illustré dans la table 9.8 :

1. Agréger indépendamment les $|C|$ S^n_{Intra} composés d'un nombre variable de valeurs en $|C|$ $S^n_{Intra} - Stat$ composés d'un nombre fini de valeurs en utilisant $|D|$ descripteurs statistiques.
2. Agréger les $|C|$ $S^n_{Intra} - Stat$ ensembles composés d'un nombre fini de valeurs obtenues à l'étape 1 en un nombre fini d'ensembles en utilisant les mêmes $|D|$ descripteurs statistiques.

Le nombre d'ensembles $S^D_{Intra} - Stat$ dépend du nombre de descripteurs statistiques choisis (un ensemble par descripteur statistique). Le nombre d'éléments $|S^D_{Intra} - Stat|$ est identique pour tous les $S^D_{Intra} - Stat$. Il dépend lui aussi du nombre de descripteurs statistiques choisis (une valeur par descripteur statistique) et est indépendant du nombre et de la taille des sous-ensembles S^n_{Intra}, qui eux dépendent du nombre de classes et du nombre de documents par classe.
Ainsi indépendamment du nombre de classes et du nombre de documents par classe, il est possible de définir la représentation intra-classe du corpus au moyen de $|D|$ ensembles de $|D|$ valeurs, où $|D|$ correspond au nombre de descripteurs statistiques choisis pour décrire le corpus.

En résumé, nous pouvons définir un corpus avec :
- un ensemble $S_{Inter} - Stat$ de $|D|$ valeurs comprises entre 0 et 1 (Section 9.2) définissant la similarité des classes entre-elles dont la taille $|S_{Inter} - Stat|$ est indépendante du nombre de classes (Section 9.3.1).
- $|D|$ ensembles $S^D_{Intra} - Stat$ composés de $|D|$ valeurs comprises entre 0 et 1 et définissant la similarité des documents au sein des classes. Le nombre d'éléments de chaque ensemble $|S_{Intra} - Stat|$ est indépendant du nombre de classes et du nombre de documents par classe (Section 9.3.2).

Pour illustrer notre propos, un exemple est présenté dans le tableau 9.8 pour D : {moyenne, variance, minimum, maximum}.
Ainsi indépendamment du nombre de classes ou de documents par classe, nous pouvons définir un corpus au moyen de $|D| \times |D| + |D|$ valeurs numériques bornées

	Moyenne	Variance	Minimum	Maximum
Classe	$S_{Inter} - Stat$			
Moyenne Variance Minimum Maximum	$S_{Intra} - Stat$			

Table 9.8: Un ensemble $S_{Inter} - Stat$ et $|D|$ ensembles $S^{D}_{Intra} - Stat$

	Moyenne	Variance	Minimum	Maximum
$S_{Inter} - Stat$	0.18	0.005	0.10	0.34
$S^{moyenne}_{Intra} - Stat$	0.19	0.02	0.03	0.45
$S^{variance}_{Intra} - Stat$	0.02	0	0	0.05
$S^{minimum}_{Intra} - Stat$	0.08	0.02	0	0.30
$S^{maximum}_{Intra} - Stat$	0.55	0.11	0.10	1

Table 9.9: Détail des |D|x|D|+|D| valeurs numériques

entre 0 et 1, où $|D|$ correspond au nombre d'opérateurs statistiques utilisés pour décrire les ensembles. Rappelons que $|D|\times|D|$ correspond au nombre d'opérations propres aux calculs intra-classes auxquels il faut ajouter les $|D|$ opérations propres aux calculs inter-classes. Un exemple est illustré dans le tableau 9.9.
Prises indépendamment les $|D| \times |D| + |D|$ valeurs ne permettent pas d'avoir une vision précise de la représentation d'un corpus mais prises ensembles, elles fournissent une description complète et cohérente du comportement des classes et de leurs compositions. Nous proposons d'utiliser ces $|D| \times |D| + |D|$ valeurs comme autant de méta-descripteurs pour décrire un corpus.

9.5 Discussion

Nous explorons une manière innovante de décrire un corpus en se basant sur une approche non encore explorée par la communauté. Cette approche basée sur la représentation des corpus et la similarité entre les différents éléments permet de résumer un corpus en un nombre fini de méta-descripteurs indépendamment du nombre de classes ou de documents du corpus. Il convient de bien définir l'ensemble D des descripteurs statistiques utilisés. En choisissant un nombre trop faible, la description ne sera pas assez précise (il est possible d'avoir deux moyennes similaires pour des réalités bien différentes), en choisissant un nombre trop grand, il y a risque de sur-apprentissage (plus la description sera détaillée et plus il sera difficile de trouver des similarités). De plus, plus le nombre de méta-descripteurs sera important, plus le temps de traitement des méta-algorithmes sera important. Enfin, il faut veiller à éviter l'utilisation de descripteurs statis-

tiques corrélés entre eux (par exemple la variance et l'écart-type) afin de ne pas introduire de redondance dans les descripteurs. Ainsi, le challenge consiste à déterminer un ensemble de statistiques D qui soit représentatif pour décrire un corpus sans risque de sur-apprentissage et sans redondance. S'il est facile d'identifier et donc d'exclure deux descripteurs statistiques redondants, il est plus délicat de définir l'ensemble D qui regroupe les caractéristiques attendues, car cela dépend in fine des ensembles à décrire. Or dans un contexte de méta-classification, l'ensemble des corpus possibles (et donc des ensembles) est par définition inconnu et infini. Des tests effectués avec notre approche rejoignent les observations réalisées dans [67], à savoir que des méta-descripteurs utiles pour prédire les performances d'un algorithme donné peuvent ne pas être utiles pour prédire les performances d'un autre algorithme.
Nous testons et discutons différents ensembles D lors des expérimentations présentées dans le chapitre suivant où nous comparons notre approche avec des méta-descripteurs de la littérature.

Chapitre

10 Expérimentations avec nos nouveaux méta-descripteurs

Nous avons dans le chapitre précédent proposé une nouvelle approche pour décrire un corpus en fonction des éléments le composant. Dans ce chapitre, nous évaluons notre proposition et nous la comparons à d'autres approches déjà éprouvées. Les approches de méta-classification demeurent avant tout des approches de classification.
Notre objectif est de pouvoir prédire le score obtenu pour un corpus donné, une mesure de performance donnée et un algorithme donné (par exemple, prédire la Micro F-mesure obtenue pour l'algorithme Cfc avec les paramètres $\alpha = 0.5$, $\beta = 0.5$ et $\omega = 0.5$ sur le corpus Itesoft). Ainsi, si nous récupérons un nouveau corpus qui serait similaire au corpus Itesoft, ce que nous pourrons évaluer en comparant les méta-descripteurs, nous pourrons prédire la Micro F-mesure obtenue pour l'algorithme Cfc avec les mêmes paramètres sur ce nouveau corpus sans avoir à exécuter l'ensemble des possibles comme nous avons procédé dans la première partie de ce manuscrit.

Sur la base de l'état de l'art, nous proposons des n-uplets <Corpus, Algorithme, Mesure, Score, Méta-descripteurs> où :
- Corpus correspond à un identifiant d'un corpus unique (par exemple le corpus *Itesoft*) ;
- Algorithme correspond à un algorithme de classification précis appliqué sur le corpus (par exemple l'algorithme SMO) ;
- Mesure correspond à la mesure utilisée pour évaluer les performances de l'algorithme précédent (par exemple la Micro F-mesure) ;
- Score correspond au score de la mesure de performance relevée pour l'algorithme donné sur le corpus (par exemple 0.608) ;

- Méta-descripteurs correspond à un ensemble de descripteurs extraits sur le corpus.

Score et méta-descripteurs sont indépendants, Score dépend du corpus, de l'algorithme et de la mesure (nous avons vu avec la première contribution que le score obtenu variait en fonction des trois éléments) alors que méta-descripteurs dépend uniquement du corpus. Corpus, algorithme, mesure, score sont des valeurs uniques alors que méta-descripteurs est en ensemble de valeurs.

Contrairement à notre première contribution, nous ne définissons pas un nouvel algorithme de classification que nous cherchons à évaluer par rapport à d'autres algorithmes, mais nous définissons de nouveaux méta-descripteurs que nous devons comparer avec d'autres types de méta-descripteurs. De plus, nous ne cherchons plus à déterminer une classe parmi un ensemble de classes, mais nous cherchons maintenant à prédire un score.
Nous sommes dans un cas de régression qui implique que :

1. Les algorithmes utilisés soient différents de ceux utilisés dans la première partie puisqu'il s'agira d'algorithmes de régression et non plus d'algorithmes de classification ;
2. L'évaluation des résultats ne sera plus une F-mesure, mais un écart constaté entre le score réel et le score prédit.

10.1 Protocole expérimental

Nous avons effectué un grand nombre d'expérimentations dans la première partie de ce manuscrit. A l'issue des expérimentations de notre première contribution, nous avions à notre disposition 36 corpus.
- le corpus *Itesoft* (Section 5.4)
- 7 corpus générés à partir de *Reuters* pour évaluer l'impact du nombre de classes (Section 5.5.1)
- 9 corpus générés à partir de *Reuters* pour évaluer l'impact du nombre de documents (Section 5.5.2)
- 11 corpus générés à partir de *Polop* pour évaluer l'impact du nombre de descripteurs (Section 5.5.3)
- 8 corpus générés à partir de *Reuters* pour évaluer l'impact du déséquilibre entre classes (Section 5.5.4)

Dans la suite de ce manuscrit, nous identifions chacun des 36 corpus par *NomCorpus_NumExpe* où :
- NomCorpus prend les valeurs *Itesoft*, *Polop* (impact du nombre de mots), *Reuters1* (impact du nombre de classes), *Reuters2* (impact du nombre de documents), *Reuters3* (impact du déséquilibre entre classes) selon la série d'expérimentations menées.

- NumExpe correspond au numéro de l'expérimentation dans une série donnée tel que défini dans les tables 5.7, 5.8, 5.9, 5.10. Par exemple, Polop_8 correspond à la $8^{ième}$ expérimentation menée sur le corpus *Polop* (tableau 5.9) et Reuters3_2 correspond à la deuxième expérimentation menée sur le corpus *Reuters* lors de l'étude de l'impact du déséquilibre (tableau 5.10). Une seule expérimentation ayant été effectuée sur le corpus *Itesoft*, l'identifiant du corpus *Itesoft* est Itesoft_1.

Pour l'évaluation de notre première contribution, chacun de ces 36 corpus a été divisé en 3 pour pouvoir effectuer une validation croisée. Nous avons donc généré 3×36 corpus soit 108 corpus. Certains sont proches (les 3 itérations génèrent des corpus différents, mais proches en terme de volume), d'autres très différents, mais chacun de ces 108 corpus est unique. Dans la suite de ce manuscrit, nous identifions chacun des 108 corpus par *NumIteration_NomCorpus_NumExpe* où *NumIteration* correspond au numéro de l'itération de validation croisée (*NumIteration* varie de 1 à 3). Par exemple 1_Polop_8 correspond au corpus généré à la première itération (sur trois) de la $8^{ième}$ expérimentation menée sur le corpus *polop*, 2_Polop_8 à la deuxième et 3_Polop_8 à la troisième.

Sur chacun de ces corpus, nous avons exécuté 128 algorithmes, mais pour des raisons de lisibilité et afin d'isoler nos deux contributions, nous ne conserverons que les 8 algorithmes de comparaison dans la suite de ces expérimentations. Pour chacun de ces 8 algorithmes de comparaison et sur chacun de ces 108 corpus, nous avons relevé 6 mesures différentes (la Micro Précision, le Micro Rappel, la Micro F-mesure ainsi que la Macro Précision, le Macro Rappel et la Macro F-mesure). Un sous-ensemble de ces mesures est donné dans le tableau 10.1 pour illustrer nos propos. Dans cet exemple, nous avons sélectionné les mesures relevées pour les corpus 1_Polop_8, 2_Polop_8 et 3_polop_8.

Pour comparer notre proposition avec celle de la littérature, nous avons cherché à prédire, pour chacun des 48 couples <algorithme de classification, mesure>, le score obtenu en construisant un modèle à partir des 108 corpus. Pour cela nous avons extrait les 108 ensembles de méta-descripteurs à partir des 108 corpus et par application d'un algorithme de régression, nous avons mesuré l'écart entre les scores prédits et les scores réels. Cet écart a été évalué avec le coefficient de corrélation (qui tend vers 1 lorsque les scores prédits sont proches des scores réels, 0 sinon). A noter qu'il n'est pas ici question de la qualité du score en lui-même, mais bien de l'écart entre le score prédit et le score réel (un fort coefficient de corrélation indique que le score prédit est proche du score observé, pas que celui-ci soit élevé). Afin d'obtenir un résultat robuste et dans la mesure où le nombre de corpus à notre disposition pour, à la fois, construire et évaluer les différents modèles de régression était limité, nous avons utilisé une validation croisée *leave-one-out* qui consiste à créer un modèle sur n-1 observations (où n correspond au nombre total d'observations), à tester sur la dernière puis à re-

Table 10.1: Ensemble des mesures relevées pour les corpus 1_Polop_8. 2_Polop_8 et 3_polop_8

Corpus	Algo	macrofscore	macroprec	macrorapp	microfscore	microprec	microrapp
1_Polop_8	ComplementNaiveBayes	0.34	0.38	0.32	0.47	0.45	0.48
1_Polop_8	DMNB	0.39	0.52	0.32	0.54	0.53	0.55
1_Polop_8	j48	0.27	0.29	0.26	0.39	0.37	0.41
1_Polop_8	LadTree	0.28	0.33	0.25	0.42	0.40	0.44
1_Polop_8	LibSVM	0.32	0.40	0.26	0.46	0.46	0.45
1_Polop_8	NaiveBayes	0.34	0.33	0.35	0.41	0.43	0.38
1_Polop_8	NaiveBayesMultinomial	0.34	0.47	0.27	0.48	0.48	0.49
1_Polop_8	SMO	0.38	0.45	0.33	0.49	0.49	0.51
2_Polop_8	ComplementNaiveBayes	0.35	0.40	0.30	0.47	0.46	0.49
2_Polop_8	DMNB	0.33	0.40	0.28	0.47	0.44	0.49
2_Polop_8	j48	0.25	0.25	0.24	0.36	0.35	0.38
2_Polop_8	LadTree	0.27	0.26	0.28	0.42	0.39	0.46
2_Polop_8	LibSVM	0.31	0.38	0.26	0.44	0.43	0.44
2_Polop_8	NaiveBayes	0.31	0.30	0.32	0.38	0.41	0.36
2_Polop_8	NaiveBayesMultinomial	0.32	0.40	0.27	0.46	0.45	0.49
2_Polop_8	SMO	0.34	0.40	0.30	0.48	0.46	0.50
3_Polop_8	ComplementNaiveBayes	0.40	0.32	0.49	0.47	0.51	0.19
3_Polop_8	DMNB	0.54	0.31	0.53	0.53	0.53	0.03
3_Polop_8	j48	0.27	0.25	0.38	0.36	0.39	0.31
3_Polop_8	LadTree	0.35	0.28	0.46	0.44	0.48	0.11
3_Polop_8	LibSVM	0.36	0.26	0.44	0.42	0.45	0.11
3_Polop_8	NaiveBayes	0.33	0.35	0.42	0.46	0.39	0.33
3_Polop_8	NaiveBayesMultinomial	0.40	0.28	0.48	0.46	0.51	0.30
3_Polop_8	SMO	0.38	0.30	0.47	0.45	0.49	0.35

produire ce test n fois (108 fois dans notre cas). Il s'agit d'un cas particulier de validation croisée où le nombre d'itérations correspond au nombre d'observations.

Pour évaluer la qualité des modèles, nous avons sélectionné un ensemble d'algorithmes de régression. Au même titre que les algorithmes de classification, nous avons testé 12 algorithmes [1] appartenant à différentes familles (SVM, Arbre de décision, K-plus-proche-voisin, Régréssion Linéaire...) afin d'éviter d'introduire des biais dans nos analyses en observant des comportements qui seraient liés aux algorithmes de régression et non à nos propositions. Pour les mêmes raisons, pour isoler les impacts de notre contribution par rapport aux éléments existants, nous avons utilisé les implémentations fournies dans Weka.

Parmi les 12 algorithmes testés, nous avons ensuite sélectionné les 5 algorithmes qui permettaient d'obtenir les meilleurs résultats sur nos données. Dans 93% des cas testés, l'un de ces 5 algorithmes suivants donne de meilleurs résultats que les 11 autres (deux approches des types plus-proches-voisins, IB2 et KStar, un arbre de décision, M5P, un algorithme de régression linéaire, LinearRegression et une approche de type SVM, SMOreg).

1. GaussianProcesses, IB1, IB2, IB3, IB4, IB5, KStar, LinearRegression, LWL, M5P, REPTree, SMOreg.

Nous venons de définir le protocole nous permettant de mesurer l'impact de différents méta-descripteurs sur nos données. Notre seconde contribution porte exclusivement sur la définition de nouveaux méta-descripteurs et il est important que seul le choix des méta-descripteurs varie afin de pouvoir mesurer l'impact de nos propositions. Il convient maintenant de discuter des méta-descripteurs utilisés, à la fois ceux que nous proposons, mais aussi ceux de la littérature avec lesquels nous nous comparons.

10.1.1 Méta-descripteurs issus de la littérature

Parmi les différentes propositions de la littérature, les approches basées sur les landmarkers reviennent souvent comme étant une proposition pertinente pour décrire un jeu de données. Pour comprendre pourquoi, il faut se rappeler que la méta-classification repose sur le principe qu'un algorithme testé sur 2 corpus identiques donne 2 résultats identiques et que 2 corpus proches donnent 2 résultats proches. Les approches basées sur les landmarkers reposent simplement sur le principe inverse, si un algorithme donne 2 résultats identiques (ou proches) sur 2 corpus, alors les corpus sont identiques (ou proches). Cela est partiellement vrai dans la mesure où un algorithme testé sur 2 corpus différents peut donner un résultat identique (ou proche), mais les approches basées sur les landmarkers supposent que deux corpus différents ne pourraient donner des résultats identiques (ou proches) indépendamment des algorithmes utilisés. Ainsi, en utilisant un ensemble d'algorithmes appartenant à différentes familles, cet effet sera limité. Nous avons décidé d'utiliser les algorithmes ComplementNaiveBayes, IB1, OneR, RepTree et ZeroR comme landmarkers qui correspondent aux critères attendus (ils sont rapides et appartiennent à différentes familles).

Nous avons mesuré la Micro F-mesure pour ces 5 algorithmes sur l'ensemble des 108 corpus. Pour illustrer, nous donnons un sous-ensemble des résultats (nous avons sélectionné 15 corpus sur les 108 pour cet exemple) dans le tableau 10.2.
Les cinq algorithmes offrent des performances très différentes. Sur nos 108 corpus, nous avons constaté une dispersion moyenne entre le plus performant et le moins performant des 5 landmarkers de 0.58 et un écart type moyen de 0.21. Dans la suite de ce manuscrit, nous ferons référence aux méta-descripteurs basés sur les landmarkers sous le nom *Land*.

Nous avons utilisé aussi un second ensemble de méta-descripteurs composé de statistiques élémentaires calculées sur les corpus : le nombre de classes, le nombre de documents, le nombre de descripteurs et nombre de descripteurs uniques. L'hétérogénéité de nos corpus (aussi bien les types, les thèmes ou la langue) nous a convaincu de ne pas utiliser la présence d'un descripteur dans une ressource extérieure comme méta-descripteurs.
Pour illustrer, nous donnons un sous-ensemble des résultats (nous avons sélec-

Table 10.2: Sous-ensemble des landmarkers calculés pour 15 de nos 108 corpus

Corpus	ComplementNaiveBayes	IB1	OneR	RepTree	ZeroR	Dispersion	Ecart Type
1_Itesoft_1	0.54	0.24	0.49	0.41	0.41	0.30	0.10
2_Itesoft_1	0.54	0.41	0.49	0.41	0.41	0.13	0.05
3_Itesoft_1	0.52	0.24	0.49	0.42	0.42	0.27	0.10
1_Polop_1	0.77	0.43	0.48	0.29	0.19	0.57	0.20
2_Polop_1	0.78	0.48	0.49	0.29	0.19	0.58	0.20
3_Polop_1	0.72	0.46	0.46	0.34	0.20	0.52	0.17
1_Polop_8	0.47	0.14	0.33	0.40	0.20	0.33	0.12
2_Polop_8	0.47	0.19	0.32	0.42	0.19	0.28	0.11
3_Polop_8	0.49	0.02	0.33	0.42	0.19	0.47	0.17
1_Reuters1_1	0.71	0.36	0.05	0.01	0.00	0.71	0.27
2_Reuters1_1	0.78	0.40	0.04	0.01	0.00	0.77	0.30
3_Reuters1_1	0.69	0.34	0.03	0.01	0.00	0.69	0.27
1_Reuters2_5	0.78	0.42	0.13	0.11	0.02	0.76	0.28
2_Reuters2_5	0.79	0.35	0.05	0.11	0.02	0.77	0.29
2_Reuters2_5	0.74	0.26	0.07	0.13	0.02	0.73	0.26

Table 10.3: Sous-ensemble des statistiques élémentaires calculées pour 15 de nos 108 corpus

Corpus	Classe	Document	Descripteur	Descripteur Unique
1_Itesoft_1	6	426	41 018	10 063
2_Itesoft_1	6	425	44 086	10 740
3_Itesoft_1	6	422	39 434	10 082
1_Polop_1	5	398	529 695	12 786
2_Polop_1	5	395	508 251	12 685
3_Polop_1	5	393	541 428	12 368
1_Polop_8	5	394	20 242	4 199
2_Polop_8	5	389	19 928	4 178
3_Polop_8	5	389	20 460	4 130
1_Reuters2_5	10	409	52 851	8 833
2_Reuters2_5	10	409	53 482	8 821
3_Reuters2_5	10	409	48 519	8 422
1_Reuters1_1	28	465	52 519	8 804
2_Reuters1_1	28	471	52 814	9 311
3_Reuters1_1	28	462	59 207	9 560

tionné les 15 mêmes corpus) dans le tableau 10.3
Dans la suite de ce manuscrit, nous ferons référence aux méta-descripteurs basés sur les statistiques élémentaires sous le nom *Stat*.

Les deux ensembles *Land* et *Stat* constituent le référentiel avec lequel nous comparons notre proposition.

10.1.2 Les nouveaux méta-descripteurs

Dans cette seconde partie, nous avons proposé une nouvelle approche permettant de définir un corpus en mesurant la similarité entre les documents ou entre les classes. Nous sommes arrivés à la conclusion qu'en choisissant $|D|$ descripteurs statistiques, nous pouvions décrire notre corpus aux travers de $|D| \times |D| + |D|$ méta-descripteurs. Pour nos expérimentations, nous avons utilisé deux ensembles de descripteurs statistiques. Le premier est composé de 14 descripteurs statis-

tiques (soit 210 méta-descripteurs), D_1={le minimum, le maximum, la moyenne, la dispersion, la variance, le 1er décile, le 2ème décile, le 3ème décile, le 4ème décile, le 5ème décile, le 6ème décile, le 7ème décile, le 8ème décile, le 9ème décile}. Dans la suite de ce manuscrit, nous ferons référence aux 210 méta-descripteurs basés sur les D_1 descripteurs statistiques sous le nom *D14*.

Pour notre second ensemble de descripteurs statistiques, nous avons décidé de supprimer les déciles (en conservant néanmoins le 5ème décile qui correspond à la médiane). Nous conservons ainsi 6 descripteurs statistiques, D_2={le minimum, le maximum, la moyenne, la dispersion, la variance, le 5ème décile}, et définissons ainsi 42 méta-descripteurs. Par analogie, nous ferons référence aux 42 méta-descripteurs basés sur les D_2 descripteurs statistiques sous le nom *D6* dans la suite de ce manuscrit.

Nous venons ainsi dans la première partie de ce chapitre de définir le protocole expérimental ainsi que 4 ensembles de méta-descripteurs que nous souhaitons évaluer et nous présentons dans la section suivante une synthèse des résultats obtenus.

10.2 Résultats

Dans un premier temps, nous avons choisi de comparer les résultats obtenus avec les 4 ensembles méta-descripteurs *Land*, *Stat*, *D14* et *D6*. Nous souhaitions savoir si nos propositions permettent d'obtenir de meilleurs résultats que les approches de la littérature retenues. Pour ce faire, nous avons évalué chacun des 48 couples <algorithmes de classification-mesures> à partir des 5 algorithmes de régression en utilisant les méta-descripteurs *Land*. Puis nous avons évalué chacun des 48 couples <algorithmes de classification-mesures> à partir des 5 algorithmes de régression en utilisant les méta-descripteurs *Stat*, puis avec les méta-descripteurs *D14* et enfin *D6*. Ainsi, pour chacun des 240 triplets[2] <algorithme de régression - algorithmes de classification - mesures >, nous obtenons 4 coefficients de corrélation. Un sous-ensemble de résultat est donné dans le tableau 10.4. Nous avons sélectionné deux algorithmes de régression (KStar et M5P), 2 algorithmes de classification (DMNB et NaiveBayes) et les 6 mesures soit un sous-ensemble de 24 résultats.

2. 48 couples x 5 algorithmes de régression

Table 10.4: Sous-ensemble des coefficients de corrélation obtenus avec différents méta-descripteurs

Algo	A	Mesure	*D14*	*D6*	*Land*	*Stat*
KStar	DMNB	macro fscore	**0.97**	0.95	0.96	0.80
KStar	DMNB	macro prec	0.96	0.92	0.94	0.77
KStar	DMNB	macro rapp	0.93	0.94	**0.95**	0.81
KStar	DMNB	micro fscore	0.97	0.93	0.94	0.73
KStar	DMNB	micro prec	0.95	0.94	0.95	0.79
KStar	DMNB	micro rapp	0.95	0.89	0.92	0.66
KStar	NaiveBayes	macro fscore	0.98	0.97	0.97	0.75
KStar	NaiveBayes	macro prec	0.98	0.98	0.97	0.72
KStar	NaiveBayes	macro rapp	0.97	0.97	0.96	0.79
KStar	NaiveBayes	micro fscore	0.98	0.96	0.96	0.73
KStar	NaiveBayes	micro prec	0.98	0.96	0.96	0.74
KStar	NaiveBayes	micro rapp	0.97	0.96	0.96	0.72
M5P	DMNB	macro fscore	0.95	0.93	0.95	0.92
M5P	DMNB	macro prec	0.93	0.93	0.94	0.90
M5P	DMNB	macro rapp	0.73	0.92	0.92	0.91
M5P	DMNB	micro fscore	0.90	0.91	0.94	0.91
M5P	DMNB	micro prec	0.92	0.91	0.94	0.92
M5P	DMNB	micro rapp	0.93	0.89	0.92	0.88
M5P	NaiveBayes	macro fscore	0.95	0.95	0.96	0.92
M5P	NaiveBayes	macro prec	0.94	**0.97**	0.96	0.90
M5P	NaiveBayes	macro rapp	0.93	0.94	0.94	0.95
M5P	NaiveBayes	micro fscore	0.94	0.90	0.93	0.95
M5P	NaiveBayes	micro prec	0.96	0.92	0.92	0.95
M5P	NaiveBayes	micro rapp	0.91	0.88	0.94	**0.95**

Sur ce sous-ensemble, nous pouvons déjà observer que selon l'algorithme de régression, l'algorithme de classification et selon la mesure évaluée, le meilleur ensemble de méta-descripteurs peut être différent comme le montrent les lignes en rouge. Par exemple, le meilleur coefficient de corrélation est obtenu avec les méta-descripteurs *D14* pour la première ligne et les méta-descripteurs *Stat* pour la dernière.
Pour synthétiser les résultats, nous avons mesuré les écarts moyens observés en comparant les ensembles de méta-descripteurs deux à deux. Nous souhaitions savoir si, sur l'ensemble des 240 observations, l'un des 4 ensembles de meta-descripteurs nous permet d'obtenir de meilleurs résultats.

Ainsi, nous avons calculé les écarts entre les ensembles *D14* et *D6* sur l'ensemble des 240 observations puis nous avons retenu la moyenne des écarts. Ensuite nous avons effectué cette comparaison pour l'ensemble de 5 autres couples (*D14* et *Land* puis *D14* et *Stat* etc.). Nous présentons les résultats dans le tableau 10.5. Nous avons indiqué en rouge l'ensemble le plus performant et en bleu lorsque l'écart entre les deux ensembles est nul).

Table 10.5: Comparaison des méta-descripteurs

Méta-descripteurs 1	Méta-descripteurs 2	Ecart Moyen
D14	***D6***	-0.04
D14	***Land***	-0.04
D14	*Stat*	0.09
D6	***Land***	0.00
D6	*Stat*	0.13
Land	*Stat*	0.14

Les résultats nous permettent d'établir une hiérarchie entre les différents ensembles. ***D6* et *Land* sont très proches sur la globalité des cas** (écart moyen à 0) et ils donnent de meilleurs résultats que les méta-descripteurs *D14* et *Stat*. Les méta-descripteurs *D14* permettent néanmoins d'obtenir de meilleurs résultats que les méta-descripteurs *Stat*.

Suite à ces résultats, nous avons voulu étudier si nous améliorerions les résultats en combinant les méta-descripteurs. Plutôt que d'utiliser un seul ensemble, nous pouvons utiliser à la fois les méta-descripteurs *D14* et *Stat* ou *D14* et *Land*. La combinaison des méta-descripteurs *D14* et *D6* n'a, en revanche, pas de sens dans la mesure où *D6* est un sous-ensemble de *D14*.
Nous avons donc dans un second temps recalculé pour les 240 triplets <algorithme de régression - algorithmes de classification - mesures >, les coefficients de corrélation obtenus en utilisant des combinaisons de méta-descripteurs. Nous présentons un sous-ensemble des résultats obtenus dans le tableau 10.6.

Table 10.6: Sous-ensemble des coefficients de corrélation obtenus en combinant deux méta-descripteurs

Algo	A	Mesure	*D14-Land*	*D14-Stat*	*D6-Land*	*D6-Stat*	*Land-Stat*
KStar	DMNB	macro fscore	0.97	0.96	0.95	0.96	0.95
KStar	DMNB	macro prec	0.96	0.96	0.91	0.94	0.95
KStar	DMNB	macro rapp	0.94	0.93	0.94	0.94	0.93
KStar	DMNB	micro fscore	0.97	0.96	0.93	0.93	0.93
KStar	DMNB	micro prec	0.95	0.94	0.94	0.95	0.95
KStar	DMNB	micro rapp	0.95	0.94	0.90	0.89	0.89
KStar	NaiveBayes	macro fscore	0.98	0.98	0.97	0.98	0.96
KStar	NaiveBayes	macro prec	0.98	0.98	0.98	0.98	0.96
KStar	NaiveBayes	macro rapp	0.97	0.97	0.97	0.97	0.95
KStar	NaiveBayes	micro fscore	0.98	0.98	0.97	0.96	0.94
KStar	NaiveBayes	micro prec	0.98	0.98	0.96	0.96	0.92
KStar	NaiveBayes	micro rapp	0.97	0.98	0.96	0.96	0.94
M5P	DMNB	macro fscore	0.95	0.93	0.96	0.96	0.96
M5P	DMNB	macro prec	0.94	0.91	0.93	0.93	0.94
M5P	DMNB	macro rapp	0.73	0.74	0.93	0.94	0.95
M5P	DMNB	micro fscore	0.93	0.87	0.93	0.91	0.94
M5P	DMNB	micro prec	0.95	0.91	0.93	0.90	0.96
M5P	DMNB	micro rapp	0.93	0.92	0.92	0.89	0.87
M5P	NaiveBayes	macro fscore	0.96	0.93	0.97	0.93	0.97
M5P	NaiveBayes	macro prec	0.95	0.95	0.97	0.96	0.97
M5P	NaiveBayes	macro rapp	0.94	0.94	0.94	0.94	0.97
M5P	NaiveBayes	micro fscore	0.95	0.94	0.97	0.95	0.97
M5P	NaiveBayes	micro prec	0.96	0.93	0.95	0.88	0.96
M5P	NaiveBayes	micro rapp	0.94	0.95	0.96	0.88	0.96

Les combinaisons de méta-descripteurs permettent d'obtenir des coefficients de corrélation de l'ordre de 90%. De la même façon que lors de notre première expérimentation, nous avons comparé les résultats deux à deux et nous comparons tout d'abord les résultats obtenus avec une combinaison de méta-descripteurs aux résultats obtenus avec un seul ensemble de méta-descripteurs dans le tableau 10.7.

Table 10.7: Comparaison des méta-descripteurs initiaux et combinés par deux

Méta-descripteurs 1	Méta-descripteurs 2	Ecart Moyen
D14-Land	*D14*	0.05
D14-Land	***Land***	0.00
D14-Stat	***D14***	0.00
D14-Stat	*Stat*	0.10
D6-Land	*D6*	0.04
D6-Land	*Land*	0.03
D6-Stat	*D6*	0.02
D6-Stat	*Stat*	0.15

Nous pouvons constater que, logiquement, prendre deux ensembles de méta-descripteurs permet de mieux décrire les corpus puisque le coefficient de corrélation moyen obtenu pour deux ensembles de méta-descripteurs est supérieur de manière significative à celui obtenu avec un seul ensemble. Cependant le gain est nul (marginalement positif) pour deux cas de figure, mais il est intéressant de noter que dans tous les cas l'introduction d'un second ensemble n'introduit pas de bruit qui dégraderait les performances.

Nous nous sommes ensuite intéressés à la comparaison des résultats obtenus avec les différentes combinaisons de deux ensembles de méta-descripteurs.

Table 10.8: Comparaison des combinaisons de deux méta-descripteurs

Méta-descripteurs 1	Méta-descripteurs 2	Ecart Moyen
D14-Land	*D14-Stat*	0.04
D14-Land	***Land-Stat***	-0.04
D14-Stat	***Land-Stat***	-0.08
D6-Land	*D6-Stat*	0.02
D6-Land	***Land-Stat***	0.00
D6-Stat	***Land-Stat***	-0.02

Les résultats nous permettent d'établir deux hiérarchies des ensembles :

1. Utiliser les deux ensembles *D14-Stat* donnent des résultats inférieurs aux deux ensembles *D14-Land* qui eux-mêmes donnent des résultats inférieurs aux deux ensembles *Land-Stat*.

2. Utiliser les deux ensembles *D6-Stat* donnent des résultats inférieurs aux deux ensembles *Land-Stat* qui donnent des résultats similaires aux deux ensembles *D6-Land*.

Ces hiérarchies confirment ce qui a été constaté avec les ensembles pris individuellement, *D6* et *Land* sont proches et permettent d'obtenir les meilleurs résultats. *D14* et *Stat* sont moins performants.

Enfin, pour compléter nos analyses, nous avons souhaité étudier le comportement en combinant les trois ensembles de méta-descripteurs (*D14-Land-Stat* puis *D6-Land-Stat*). Un sous-ensemble des résultats est présenté dans le tableau 10.9.

Table 10.9: Sous-ensemble des coefficients de corrélation obtenus en combinant trois méta-descripteurs

Algo	A	Mesure	*D14-Land-Stat*	*D6-Land-Stat*
KStar	DMNB	macro fscore	0.95	0.96
KStar	DMNB	macro prec	0.94	0.94
KStar	DMNB	macro rapp	0.77	0.95
KStar	DMNB	micro fscore	0.92	0.93
KStar	DMNB	micro prec	0.95	0.94
KStar	DMNB	micro rapp	0.92	0.92
KStar	NaiveBayes	macro fscore	0.96	0.98
KStar	NaiveBayes	macro prec	0.95	0.97
KStar	NaiveBayes	macro rapp	0.95	0.94
KStar	NaiveBayes	micro fscore	0.96	0.96
KStar	NaiveBayes	micro prec	0.93	0.94
KStar	NaiveBayes	micro rapp	0.95	0.96
M5P	DMNB	macro fscore	0.96	0.96
M5P	DMNB	macro prec	0.96	0.94
M5P	DMNB	macro rapp	0.93	0.94
M5P	DMNB	micro fscore	0.96	0.93
M5P	DMNB	micro prec	0.94	0.95
M5P	DMNB	micro rapp	0.94	0.89
M5P	NaiveBayes	macro fscore	0.98	0.98
M5P	NaiveBayes	macro prec	0.98	0.98
M5P	NaiveBayes	macro rapp	0.97	0.97
M5P	NaiveBayes	micro fscore	0.98	0.96
M5P	NaiveBayes	micro prec	0.98	0.96
M5P	NaiveBayes	micro rapp	0.98	0.96

Prendre trois ensembles de méta-descripteurs nous permet d'obtenir des coefficients de corrélation moyens de 94%, ce qui confirme une amélioration globale par rapport aux ensembles doubles (92%) ou aux ensembles uniques (86%).
Nous présentons dans le tableau 10.10, les résultats obtenus avec trois ensembles comme méta-descripteurs au regard des résultats mesurés avec les autres combinaisons d'ensembles de méta-descripteurs testés précédemment.

Table 10.10: Comparaison des méta-descripteurs initiaux et combinés par trois

Méta-descripteurs 1	Méta-descripteurs 2	Ecart Moyen
D14-Land-Stat	*D14*	0.048
D14-Land-Stat	*Land*	0.005
D14-Land-Stat	*Stat*	0.140
D14-Land-Stat	*D14-Land*	0.002
D14-Land-Stat	*D14-Stat*	0.044
D14-Land-Stat	***Land-Stat***	-0.033
D6-Land-Stat	*D6*	0.044
D6-Land-Stat	*Land*	0.039
D6-Land-Stat	*Stat*	0.174
D6-Land-Stat	*Land-Stat*	0.002
D6-Land-Stat	*D6-Land*	0.005
D6-Land-Stat	*D6-Stat*	0.023

Nous pouvons constater que prendre trois ensembles de méta-descripteurs ne permet pas forcément de mieux décrire les corpus puisqu'utiliser les ensembles *D14-Land-Stat* donnent des résultats de moindre qualité qu'en utilisant simplement les ensembles *Land-Stat*.
En revanche, les méta-descripteurs *D6* nous permettent d'obtenir une amélioration des performances par rapport à toutes les autres combinaisons de méta-descripteurs étudiés. Nous résumons dans le tableau 10.11 les coefficients de corrélation moyens constatés pour illustrer notre propos.

Table 10.11: Résumé des coefficients de corrélation moyens mesurés

Ensemble Unique	
D14	0.866
D6	0.905
Land	0.910
Stat	0.775
Moyenne	0.864
Deux ensembles	
D14-Land	0.912
D14-Stat	0.870
D6-Land	0.944
D6-Stat	0.925
Land-Stat	0.947
Moyenne	0.920
Trois ensembles	
D14-Land-Stat	0.914
D6-Land-Stat	0.949
Moyenne	0.937

Ces expérimentations nous permettent de conclure que l'utilisation de notre proposition est une alternative crédible aux méta-descripteurs existants et nous discutons dans le chapitre suivant des avantages et inconvénients des différents ensembles de méta-descripteurs.

Chapitre

11

Discussions et conclusions

Pour notre seconde contribution, nous avons donc proposé une nouvelle approche permettant de résumer un corpus en un nombre fini de méta-descripteurs indépendamment du nombre de classes ou de documents par classes, indépendamment du type de documents du corpus, de la langue utilisée. Nos nouveaux méta-descripteurs nous permettent d'obtenir des résultats tout à fait comparables aux approches de la littérature et possèdent des propriétés intéressantes. L'approche proposée permet de définir un grand nombre de méta-descripteurs à partir d'un nombre restreint de descripteurs statistiques (pour |D| descripteurs nous définissons, $|D| \times (|D|+1)$ méta-descripteurs). Il convient simplement de trouver le juste équilibre pour éviter les effets de sur-apprentissage ou de sous-apprentissage. Par exemple, dans nos expérimentations, l'ensemble de 6 descripteurs statistiques *D6* nous permet d'obtenir de meilleurs résultats que l'ensemble de 14 descripteurs statistiques *D14*.

Son implémentation est aisée puisque notre approche repose sur le calcul de distance entre deux documents (ou classes) puis sur l'utilisation de fonctions statistiques relativement simples.
Nos méta-descripteurs permettent de comprendre et de visualiser le type de corpus traité (ai-je des classes très similaires ou au contraire très différentes ? les documents au sein de mes classes sont-ils homogènes ou hétérogènes ?). Cette connaissance peut être affinée lors du processus d'extraction des méta-descripteurs. En effet, lors des étapes intermédiaires, nous obtenons une vision extrêmement détaillée du comportement de chaque classe avant de les agréger pour obtenir un nombre fini de méta-descripteurs. Cette analyse peut permettre de détecter des comportements anormaux (pourquoi ai-je un document qui se retrouve à l'écart

de la majorité des documents de ma classe ? Pourquoi deux classes sont elles si similaires alors qu'à priori rien ne le prédisposait ?) ou de modifier le corpus en conséquence (ne serait-il pas préférable de regrouper deux classes trop similaires ? ou au contraire scinder une classe en deux ?)

Notre approche n'implique pas de partitionner les corpus comme c'est le cas avec les approches fondées sur les landmarkers, ce qui nous garantit l'absence de biais introduit par l'utilisation de sous-corpus. De plus, notre approche permet de séparer les problématiques de classification des problématiques de description là ou les approches fondées sur les landmarkers s'appuient sur des problématiques de classification pour traiter une problématique de description.

Nos expérimentations ont montré que l'utilisation de statistiques simples ne permettait pas d'obtenir des résultats satisfaisants. En effet, il parait logique que le nombre de documents ou de classes ne puisse suffire à décrire un corpus avec un maximum d'exhaustivité surtout lorsque les différents corpus sont statistiquement proches (nombre similaire de classes, de documents, de descripteurs).

Chapitre

12

Conclusion générale

12.1 Synthèse des contributions

Nous nous positionnons pour commencer dans un contexte de classification de documents pour traiter des corpus composés d'un petit volume de données. Dans la première partie de cette thèse, nous avons présenté et expérimenté une méthode de pondération adaptée au traitement de ce type de corpus. Notre pondération est particulièrement adaptée lorsque le nombre de descripteurs est réduit tout en offrant de bonnes performances lorsque le nombre de classes ou de documents est faible. Nous avons rendu cette pondération paramétrable afin de pouvoir s'adapter à la composition des corpus. Pouvoir prendre en compte les différentes caractéristiques d'un corpus (nombre de classes, nombre de documents, nombres de descripteurs) permet d'améliorer les performances des algorithmes de classifications. Nos expérimentations démontrent l'intérêt de rendre paramétrable notre approche tout en levant une nouvelle problématique : comment déterminer les meilleurs paramètres pour traiter un problème donné ? Nous répondons à cette question dans la seconde partie de cette thèse.

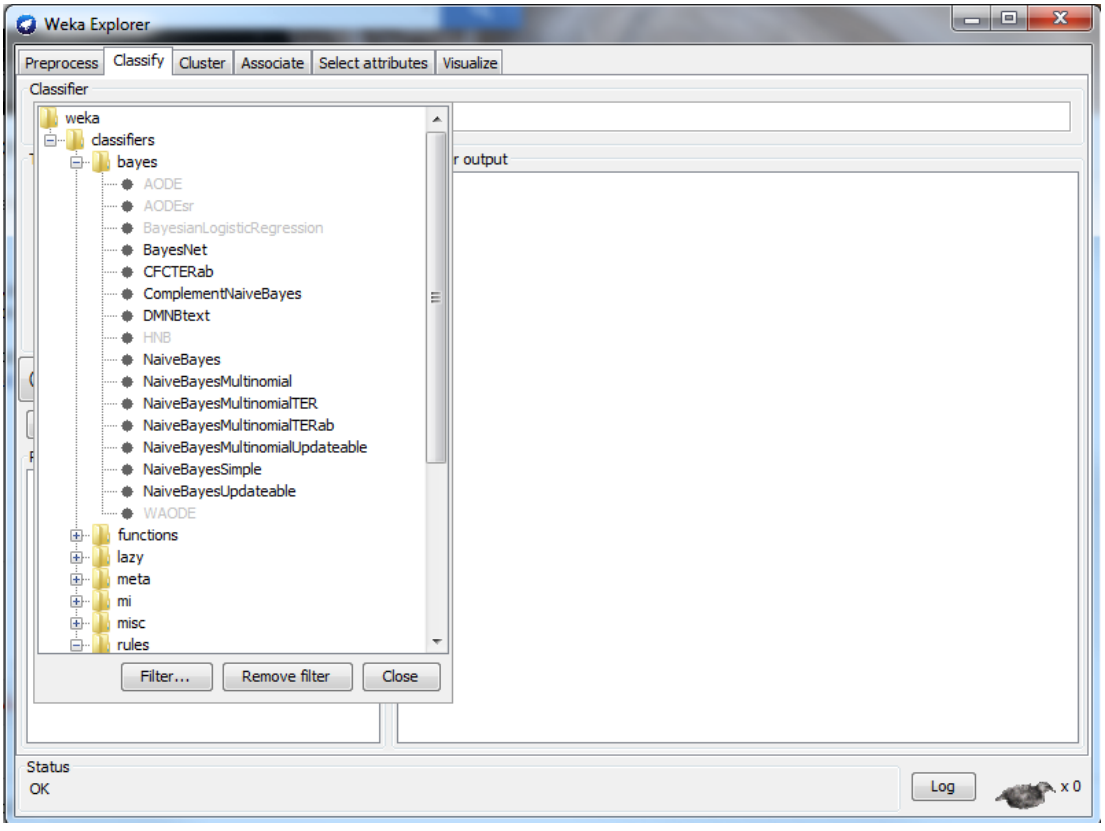
Pour notre seconde contribution, nous nous positionnons dans un contexte différent, celui de la méta-classification et plus précisément de la méta-description. Nous proposons une approche originale permettant de décrire un corpus indépendamment du nombre de classes ou de documents par classes, indépendamment du type de documents du corpus et de la langue utilisée. Pour cela, nous proposons de mesurer la similarité entre les classes et les documents puis de les agréger afin de définir une "empreinte" du corpus. Cette "empreinte" sera comparée avec les "empreintes" d'autres corpus pour lesquels les paramètres optimaux sont connus. Les

expérimentations menées confirment que nos nouveaux méta-descripteurs sont une alternative intéressante au regard des solutions actuellement proposées dans la littérature.

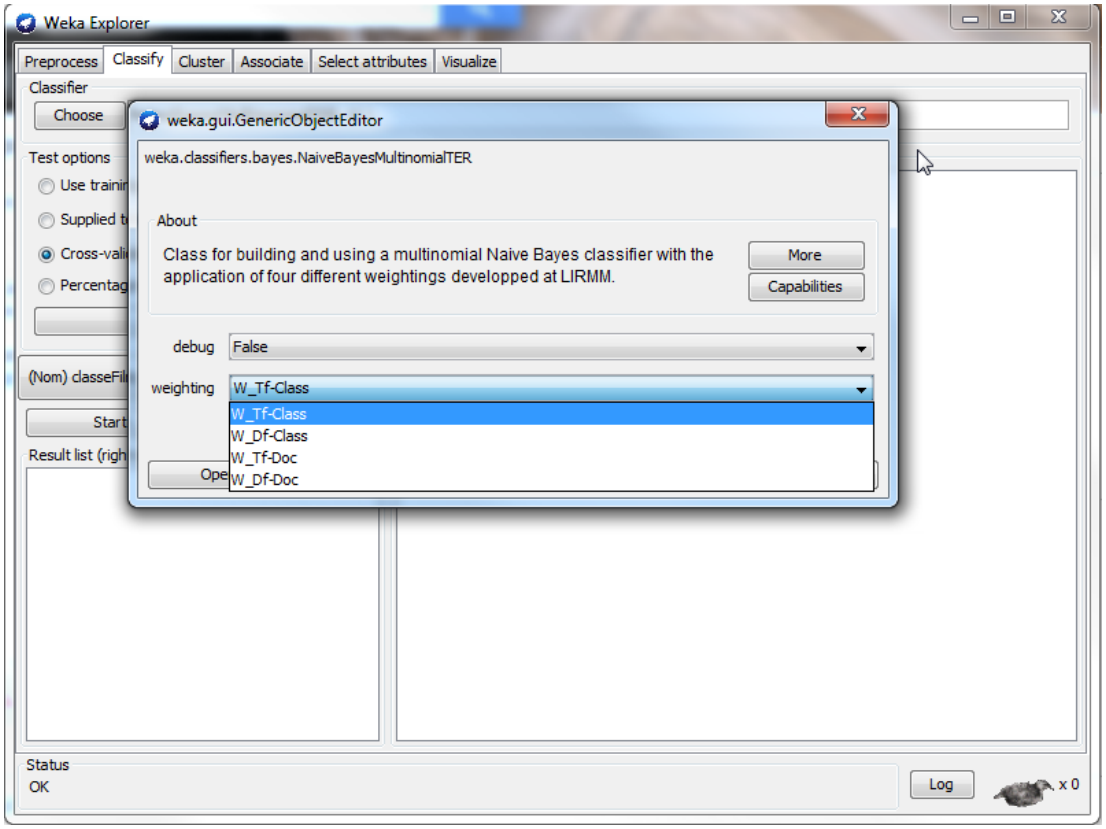
Les travaux présentés dans ce manuscrit ont fait l'objet de plusieurs prototypes et d'une intégration industrielle que nous présentons dans la section suivante.

12.2 Prototypes

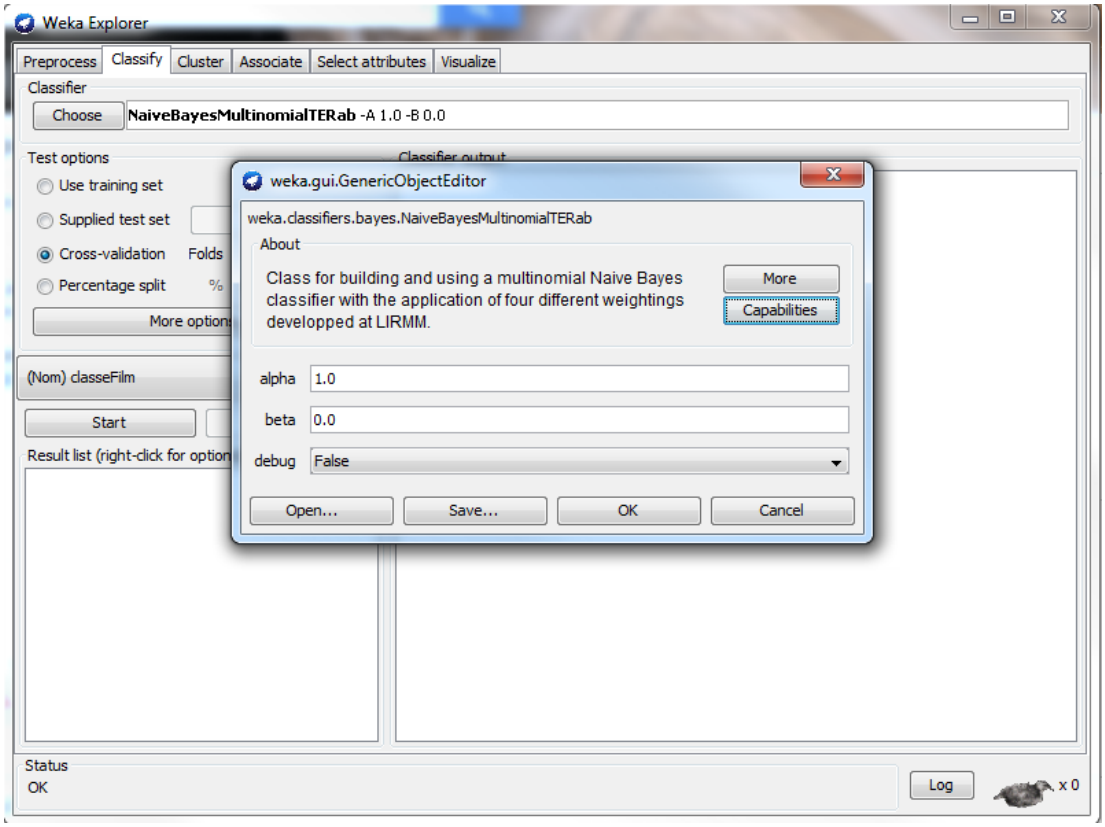
Trois différentes implémentations ont été réalisées. Pour la première, l'objectif était de rendre notre proposition accessible à travers différentes interfaces du logiciel WEKA. Dans l'interface graphique de Weka Explorer, nous pouvons trouver ces algorithmes dans l'onglet *Classify* :



La pondération de *NaiveBayesMultinomialTER* se choisit à l'aide d'un menu déroulant :



Les valeurs des paramètres sont configurables via les options comme habituellement dans l'environnement Weka :



L'intégration dans Weka permet d'utiliser les fonctionnalités connexes déjà présentes dans l'environnement WEKA par exemple les filtres, les outils de sélection, les processus de validation ou encore l'évaluation et la visualisation des résultats.

La seconde implémentation avait pour objectif le développement d'un prototype en C++ qui devait servir de base pour l'intégration de nos approches dans les logiciels d'ITESOFT. Ce développement est composé de la manière suivante :

- Un prototype fonctionnel portant l'ensemble des travaux présentés dans ce manuscrit.
- Une Dll réalisée en partenariat avec la société Itesoft prête à être intégrée dans les logiciels Itesoft

Le prototype est composé de plusieurs modules :

- Une interface graphique permettant la sélection de fichiers et de paramètres (cf. Figure 12.1) ;
- Un module de validation croisée ;
- Un module d'évaluation des performances et de présentation des résultats (cf. Figure 12.2) ;
- Un module de pondération comprenant les cinq paramètres pour l'approche

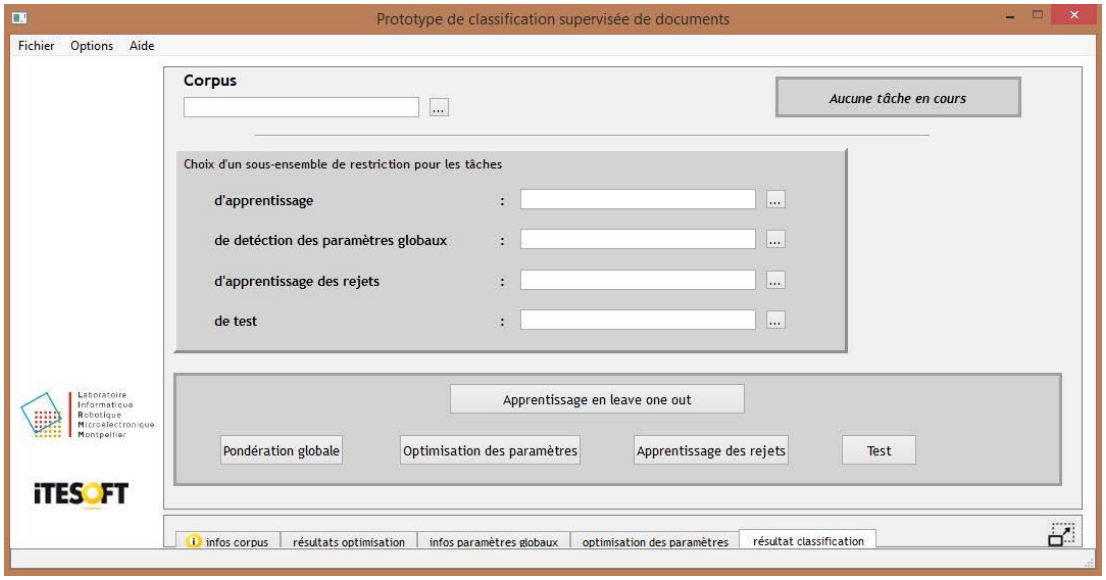


Figure 12.1: Prototype - Interface principale du prototype

NaiveBayes [1] ;
- Un module de détection automatique du couple paramètres/approche le plus performant (cf. Figure 12.3).

Suite à ce prototype, qui fut présenté à Itesoft lors des 2^{nd} Innovation days d'ITESOFT [2], une Dll a été développée pour être intégrée dans le logiciel d'Itesoft.

Ces différents développements, ainsi que les démonstrations qui ont pu être effectués, nous ont permis de définir des axes d'amélioration et des pistes de recherche qui sont détaillées dans la section suivante.

12.3 Perspectives de recherche

Premièrement, concernant notre pondération, nous pensons qu'elle peut être utile dans un contexte de "*Feature Selection*". En effet, à chaque descripteur du corpus est affecté un score et nous pensons qu'utiliser ce score peut permettre de ne conserver que les descripteurs les plus pertinents. Deux solutions ont été envisagées. Tout d'abord nous pourrions n'utiliser que les "k" descripteurs les plus pertinents de chaque classe, par exemple les 10 descripteurs ayant un score le plus élevé de chaque classe. Le problème étant alors de définir le nombre "k" le plus pertinent. Il est aussi possible que le "$k^{ème}$" descripteur d'une classe ait un score beaucoup plus faible que le "$k^{ème}$" descripteur d'une autre classe. La seconde solution consiste à utiliser un seuil au-delà duquel seraient conservés les descripteurs,

1. Pour le prototype, l'approche NaivesBayes a été préférée à l'approche CFC
2. workshop international regroupant les partenaires européens de la société

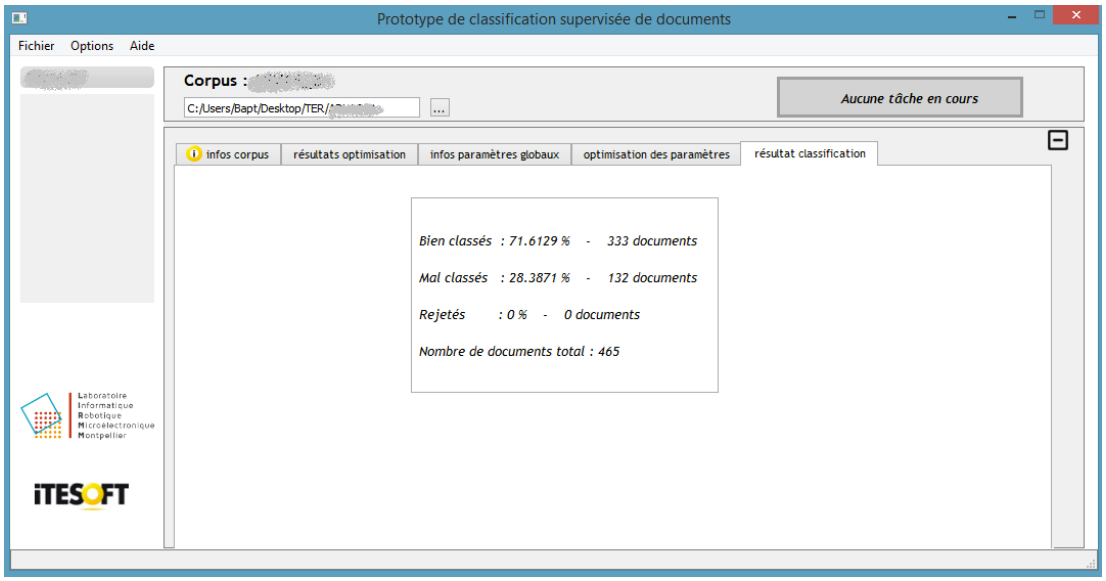


Figure 12.2: Prototype - Restitution des résultats

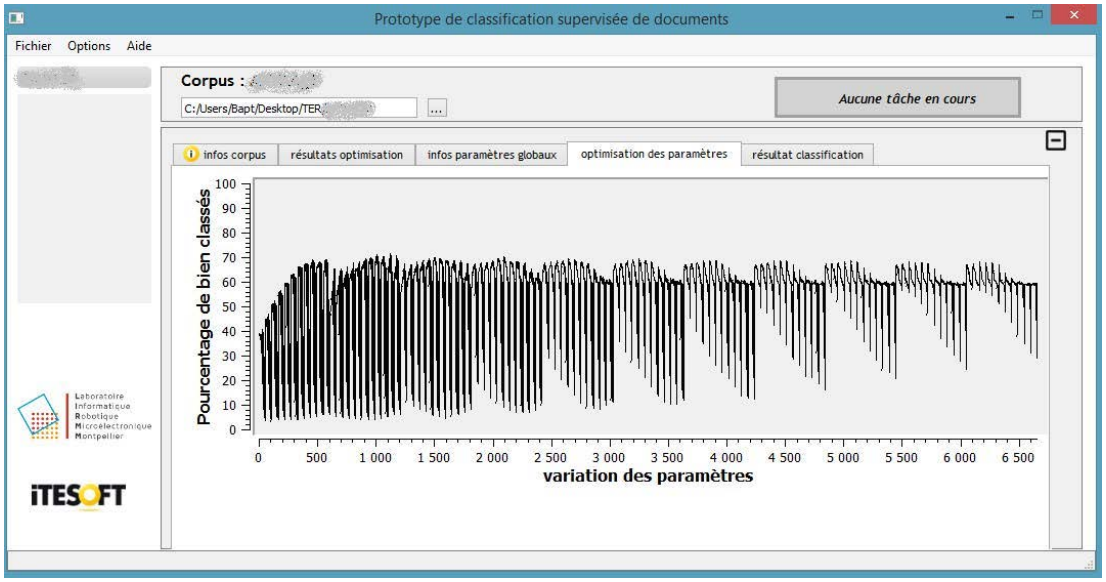


Figure 12.3: Prototype - Détection des paramètres

la problématique étant alors de définir un tel seuil. Des expérimentations préliminaires ont montré que le nombre "k" ou le seuil qui maximisent les performances varient d'un corpus à l'autre.

Deuxièmement, dans un contexte industriel, il est souvent préférable de ne pas classer plutôt que de mal classer (autrement dit, la Précision est plus importante que le Rappel). Pour ce faire, un processus de rejet peut être utilisé. Dans nos approches, la décision finale de l'affectation à une classe est définie grâce à un score. Pour un document à classer, nous attribuons un score pour chacune des classes candidates et nous choisissons la classe ayant le score maximum. La solution la plus communément utilisée est la définition d'un seuil au-dessous duquel un document ne sera pas affecté. En fixant un seuil élevé, nous allons améliorer la précision, mais nous allons aussi rejeter un ensemble de documents dont le système connaissait pourtant l'affectation. Le défi consiste alors à rejeter le maximum de documents pour qui le système se trompe tout en rejetant le minimum de documents pour qui le système ne se trompe pas. Les travaux présentés dans cette thèse reposent sur la notion d'hétérogénéité des corpus. Nous avons tout au long de ce manuscrit mis en évidence que deux corpus différents devaient être traités différemment. Nous pensons que la définition d'un seuil de rejet doit obéir à cette logique de spécificité des corpus. Autrement dit, il n'existe pas un seuil qui soit indépendant du corpus. De plus, nous pensons qu'il est possible d'utiliser les scores des deux classes les plus probables pour améliorer cette fiabilité. Intuitivement, si deux classes obtiennent un score très proche, le risque que le système se trompe est plus élevé que si l'une des deux obtient un score bien supérieur à l'autre. En revanche cette notion d'écart minimum entre classes doit aussi dépendre des classes du corpus. Par exemple, il est logique qu'un document ait un score très proche pour deux classes très similaires (par exemple "Chat" et "Chien") alors que l'on s'attend à trouver un écart plus important si les deux classes sont très différentes (par exemple "Chat" et "Avion"). Ainsi, à écart identique, la probabilité de se tromper varie selon la similarité des classes. Nous pensons que la similarité des scores doit être corrélée à la similarité des classes et qu'étudier ces aspects permettrait d'améliorer la précision de notre approche.

Troisièmement, nous n'avons pas trouvé trace dans la littérature de l'utilisation de la méta-classification pour détecter la meilleure représentation des données et répondre aux questions : faut-il extraire les lemmes ou les racines ? Faut-il utiliser des ngrammes de lettres ou des ngrammes de descripteurs ? Et si oui, faut-il privilégier des bigrammes, des trigrammes...
Au même titre qu'il n'existe pas d'algorithmes plus performants que les autres indépendamment du corpus, il n'existe pas une représentation donnée qui permettrait d'obtenir de meilleurs résultats quelque soit le corpus étudié. Par exemple, la lemmatisation pourtant reconnue comme une bonne pratique, s'avère être un frein dès lors que la conjugaison, le genre ou le nombre s'avèrent discrimants pour différencier les classes. De plus, la lemmatisation repose sur des modèles

grammaticaux et des dictionnaires et il n'est pas toujours évident d'extraire les lemmes lorsque les textes ne correspondent pas aux modèles (par exemple en cas de texte bruité ou à la syntaxe particulière comme les SMS ou les tweets). Avec une approche de type "sac de mots", les performances des algorithmes de classification sont indépendantes du contenu linguistique des documents, mais dépendent uniquement de leurs représentations vectorielles. Nos méta-descripteurs, en agrégeant des similarités, sont indépendants du type de descripteurs. Si nous avons dans ce manuscrit calculé la similarité en se basant sur le "terme", nous aurions pu mesurer la similarité en considérant des "ngrammes de lettres", des "lemmes" ou des "racines". Ainsi pour un même corpus nous pourrions calculer son "empreinte" avec les termes puis son "empreinte" avec des bigrammes de mots et, en comparant ces deux "empreintes" avec les exemples passés (indépendamment du type de descripteurs utilisé dans ces expériences), notre intuition est que nous pourrions détecter la meilleure représentation du corpus.

Enfin, pour conclure, nous pensons qu'il serait intéressant d'évaluer l'intérêt et l'applicabilité des nouveaux méta-descripteurs dans le domaine des règles d'association. En effet l'extraction de règles d'association est l'une des techniques de fouilles de données la plus utilisée dans le domaine de l'extraction de connaissances. Il existe aujourd'hui de très nombreuses propositions pour extraire ces règles d'un gros volume de données et obtenir différentes mesures : support, confiance, lift, etc. Cependant en fonction des données manipulées, les comportements de ces algorithmes sont très différents. Par exemple, un algorithme de type Apriori sera très efficace dans le cas de matrices creuses et complètement inefficace dans le cas de données denses. Inversement un algorithme comme CloseSt aura un comportement complètement opposé. Étant donnés les temps d'extraction, connaître à l'avance le type d'algorithme à utiliser pourrait être très utile pour le décideur. Ainsi nous pensons qu'il serait intéressant d'étudier et d'expérimenter l'impact de nouveaux méta-descripteurs pour l'identification des meilleurs algorithmes d'extraction de règles selon les types de données.

Liste des publications

1. **F. Bouillot**, P. Poncelet et M. Roche. "*Mesurer la proximité entre corpus par de nouveaux méta-descripteurs*". Dans les actes de la conférence francophone en Recherche d'Information et Applications (CORIA 2015), To appear, Paris, France, Mars 2015.

2. **F. Bouillot**, P. Poncelet et M. Roche. "*Classification of Small Datasets : Why Using Class-Based Weighting Measures ?*". In Proceedings of the 21st International Symposium on Methodologies for Intelligent Systems (ISMIS 2014), LNCS, Springer, Roksilde, Danemark, juin 2014.

3. **F. Bouillot**, P. Poncelet et M. Roche. "*De nouvelles pondérations adaptées à la classification de petits volumes de données textuelles*". Actes des 14ièmes Journées Francophone "Extraction et Gestion des Connaissances" (EGC 2014), Rennes, France, janvier 2014.

4. **F. Bouillot**, P. Nhat Hai, N. Béchet, S. Bringay, D. Ienco, S. Matwin, P. Poncelet, M. Roche, M. Teisseire. "*How to Extract Relevant Knowledge from Tweets ?*" In Springer CCSI (Communications in Computer and Information Science), post proceedings of ISIP ?2012 (International Workshop on Information Search, Integration and Personalization), 2013.

5. **F. Bouillot**, O. Gout, P. Magnier, C. Pénin, P. Poncelet, M. Roche. "*Vers un outil de cartographie : qui est l'expert ?*". Démonstration, Actes des 13ièmes Journées Francophone "Extraction et Gestion des Connaissances"

(EGC 2013), Démo paper, Toulouse, France, janvier 2013.

6. **F. Bouillot**, P. Poncelet, M. Roche, D. Ienco, S. Matwin and E. Bigdeli. "*French Presidential Elections : What are the Most Efficient Measures for Tweets ?*". In Proceedings of PLEAD'12 Workshop (Politics, Elections and Data) - 21st ACM International Conference on Information and Knowledge Management, Maui Hawaii, USA, 2012.

7. **F. Bouillot**, P. Poncelet and M. Roche. "*How and Why Exploit Tweet's Location Information ?*". In Proceedings of the 15th International Conference on Geographic Information Science (AGILE'12), Avignon, France, 2012.

8. S. Bringay, N. Béchet, **F. Bouillot**, P. Poncelet, M. Roche and M. Teisseire. "*Towards an On-Line Analysis of Tweets Processing*". In Proceedings of the 22nd International Conference on Database and Expert Systems Applications (DEXA 2011), LNCS, Springer, Toulouse, France, August 2011.

9. S. Bringay, N. Béchet, **F. Bouillot**, P. Poncelet, M. Roche et M. Teisseire. "*Analyse de gazouillis en ligne*". Actes 7ièmes Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2011), Clermont-Ferrand, Juin 2011.

Bibliographie

[1] D. W. Aha, « Generalizing from case studies : a case study », dans *In Proceedings of the Ninth International Conference on Machine Learning*, 1992, p. 1–10.

[2] A. Aizawa, « An information-theoretic perspective of tf&mdash ;idf measures », dans *Inf. Process. Manage.*, t. 39, jan. 2003, p. 45–65.

[3] S. M. Ali et S. D. Silvey, « A General Class of Coefficients of Divergence of One Distribution from Another », dans *Journal of the Royal Statistical Society. Series B (Methodological)*, t. 28, 1966, p. 131–142.

[4] S. Ali et K. A. Smith-Miles, « A meta-learning approach to automatic kernel selection for support vector machines », dans *Neurocomputing*, t. 70, 2006, p. 173–186.

[5] S. Ali et K. Smith-Miles, « On optimal degree selection for polynomial kernel with support vector machines : theoretical and empirical investigations », dans *Int. J. Know.-Based Intell. Eng. Syst.*, t. 11, jan. 2007, p. 1–18.

[6] S. Ali et K. A. Smith, « Matching svm kernel's suitability to data characteristics using tree by fuzzy c-means clustering », dans *Design and application of hybrid intelligent systems*, IOS Press, 2003, p. 553–562.

[7] S. Ali et K. A. Smith, « On learning algorithm selection for classification », dans *Appl. Soft Comput.*, t. 6, jan. 2006, p. 119–138.

[8] R. Basili, M. Cammisa, A. Moschitti et I Rome, « A semantic kernel to classify texts with very few training examples », dans *Informatica (Slovenia)*, t. 30, 2006, p. 163–172.

[9] J. Baxter, « A bayesian/information theoretic model of learning to learn viamultiple task sampling », dans *Mach. Learn.*, t. 28, juil. 1997, p. 7–39.

[10] H. Bensusan et C. Giraud-Carrier, « Casa batla is in passeig de gracia or how landmark performances can describe tasks », dans *Proceedings of the ECML-00 Workshop on Meta-Learning : Building Automatic Advice Strategies for Model Selection and Method Combination*, 2000, p. 29–46.

[11] H. Bensusan, « God doesn't always shave with occam's razor - learning when and how to prune », dans *Proceedings of the 10th European Conference on Machine Learning*, 1998, p. 119–124.

[12] H. Bensusan et C. G. Giraud-Carrier, « Discovering task neighbourhoods through landmark learning performances », dans *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 2000, p. 325–330.

[13] H. Bensusan et A. Kalousis, « Estimating the predictive accuracy of a classifier », dans *Proceedings of the 12th European Conference on Machine Learning*, 2001, p. 25–36.

[14] H. Berrer, I. Paterson et J. Keller, « Evaluation of machine-learning algorithm ranking advisors », dans *In Proceedings of the PKDD-2000 Workshop on DataMining, Decision Support, Meta-Learning and ILP : Forum for Practical Problem Presentation and Prospective Solutions*, 2000.

[15] N. Bhatt, A. Thakkar et A. Ganatra, « A survey and current research challenges in meta learning approaches based on dataset characteristics », dans *International Journal of Soft Computing and Engineering*, t. 2, 2012, p. 234–247.

[16] N. Bhatt, A. Thakkar, A. Ganatra et N. Bhatt, « Ranking of classifiers based on dataset characteristics using active meta learning », dans *International Journal of Computer Applications*, t. 69, 2013, p. 31–36.

[17] P. B. Brazdil, C. Soares et J. P. Da Costa, « Ranking learning algorithms : using ibl and meta-learning on accuracy and time results », dans *Mach. Learn.*, t. 50, mar. 2003, p. 251–277.

[18] P. Brazdil, C. G. Giraud-Carrier, C. Soares et R. Vilalta, *Metalearning - Applications to Data Mining*. Springer, 2009.

[19] C. E. Brodley, « Recursive automatic bias selection for classifier construction », dans *Mach. Learn.*, t. 20, juil. 1995, p. 63–94.

[20] C. E. Brodley et P. Smyth, « Applying classification algorithms in practice », dans *Statistics and Computing*, t. 7, jan. 1997, p. 45–56.

[21] C. Buckley, « Automatic query expansion using smart : trec 3 », dans *In Proceedings of The third Text REtrieval Conference (TREC-3)*, 1994, p. 69–80.

[22] Z. Cataltepe et E. Aygun, « An improvement of centroid-based classification algorithm for text classification », dans *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, 2007, p. 952–956.

[23] A. Cayci, S. Eibe, E. Menasalvas et Y. Saygin, « Bayesian networks to predict data mining algorithm behavior in ubiquitous computing environments », dans *Proceedings of the 2010 International Conference on Analysis of Social Media and Ubiquitous Data*, 2011, p. 119–141.

[24] S.-H. Cha, « Comprehensive survey on distance/similarity measures between probability density functions », dans *International Journal of Mathematical Models and Methods in Applied Sciences*, t. 1, 2007, p. 300–307.

[25] C.-C. Chang et C.-J. Lin, « Libsvm : a library for support vector machines », dans *ACM Transactions on Intelligent Systems and Technology (TIST)*, t. 2, 2011, p. 27.

[26] O. Chapelle, B. Schölkopf, A. Zien et al., *Semi-supervised learning*. MIT press Cambridge, 2006, t. 2.

[27] O. Chapelle, V. Vapnik, O. Bousquet et S. Mukherjee, « Choosing multiple parameters for support vector machines », dans *Machine learning*, t. 46, 2002, p. 131–159.

[28] M. Chen, X. Jin et D. Shen, « Short text classification improved by learning multi-granularity topics », dans *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, 2011, p. 1776–1781.

[29] W. T. Chuang, A. Tiyyagura, J. Yang et G. Giuffrida, « A fast algorithm for hierarchical text classification », dans *In DaWaK 2000 : Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, 2000, p. 409–418.

[30] D. A. Cohn, Z. Ghahramani et M. I. Jordan, « Active learning with statistical models », dans *arXiv preprint cs/9603104*, 1996.

[31] N. Cristianini, C. Campbell et J. Shawe-taylor, « Dynamically adapting kernels in support vector machines », dans *Advances in Neural Information Processing Systems 11*, 1998, p. 204–210.

[32] N. Cristianini et J. Shawe-Taylor, *An Introduction to Support Vector Machines : And Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[33] F. Debole et F. Sebastiani, « Supervised term weighting for automated text categorization », dans *Proceedings of the 2003 ACM Symposium on Applied Computing*, 2003, p. 784–788.

[34] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer et R. Harshman, « Indexing by latent semantic analysis », dans *Journal of the American Society for Information Science (JASIS)*, t. 41, 1990, p. 391–407.

[35] Z.-H. Deng, S.-W. Tang, D.-Q. Yang, M.-Y. Li et K.-Q. Xie, « A comparative study on feature weight in text categorization », dans *Advanced Web Technologies and Applications*, t. 3007, 2004, p. 588–597.

[36] Z.-H. Deng, S.-W. Tang, D.-Q. Yang, M. Zhang, X.-B. Wu et M. Yang, « A linear text classification algorithm based on category relevance factors », English, dans *Digital Libraries : People, Knowledge, and Technology*, t. 2555, 2002, p. 88–98.

[37] R. Engels et C. Theusinger, « Using a data metric for preprocessing advice for data mining applications. », dans *ECAI*, 1998, p. 430–434.

[38] J. R. Firth, « A synopsis of linguistic theory 1930-55. », dans *Studies in Linguistic Analysis (special volume of the Philological Society)*, t. 1952-59, 1957, p. 1–32.

[39] G. Forman et I. Cohen, « Learning from little : comparison of classifiers given little training », dans *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2004, p. 161–172.

[40] J. V. Frasch, A. Lodwich, F. Shafait et T. M. Breuel, « A bayes-true data generator for evaluation of supervised and unsupervised learning methods », dans *Pattern Recogn. Lett.*, t. 32, août 2011, p. 1523–1531.

[41] K. Furdík, J. Paralič et G. Tutoky, « Meta-learning method for automatic selection of algorithms for text classification », dans *Proc. of the Central European Conference on Information and Intelligent Systems (CECIIS 2008)*, 2008, p. 24–26.

[42] G. W. Furnas, T. K. Landauer, L. M. Gomez et S. T. Dumais, « Human factors in computer systems », dans, 1984, chap. Statistical Semantics : Analysis of the Potential Performance of Keyword Information Systems, p. 187–242.

[43] J. Furnkranz et J. Petrak, « An evaluation of landmarking variants », dans *Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2001)*, 2001, p. 57–68.

[44] J. Gama et P. Brazdil, « Characterization of classification algorithms », dans *Proceedings of the 7th Portuguese Conference on Artificial Intelligence : Progress in Artificial Intelligence*, 1995, p. 189–200.

[45] C. Giraud-Carrier, « Metalearning-a tutorial », dans *Tutorial at the 2008 International Conference on Machine Learning and Applications, ICMLA*, 2008.

[46] C. Giraud-Carrier, R. Vilalta et P. Brazdil, « Introduction to the special issue on meta-learning », dans *Machine Learning*, t. 54, 2004, p. 187–193.

[47] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st. Addison-Wesley Longman Publishing Co., Inc., 1989.

[48] T. A. F. Gomes, R. B. C. Prudêncio, C. Soares, A. L. D. Rossi et A. C. P. L. F. Carvalho, « Combining meta-learning and search techniques to select parameters for support vector machines », 1, t. 75, 2012, p. 3–13.

[49] H. Guan, J. Zhou et M. Guo, « A class-feature-centroid classifier for text categorization », dans *Proceedings of the 18th international conference on World wide web*, 2009, p. 201–210.

[50] E.-H. Han et G. Karypis, « Centroid-based document classification : analysis and experimental results », dans *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 2000, p. 424–431.

[51] E.-H. Han, G. Karypis et V. Kumar, « Text categorization using weight adjusted k-nearest neighbor classification », dans *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2001, p. 53–65.

[52] T. Hertz, A. B. Hillel et D. Weinshall, « Learning a kernel function for classification with small training samples », dans *Proceedings of the 23rd International Conference on Machine Learning*, 2006, p. 401–408.

[53] M. Hilario, P. Nguyen, H. Do, A. Woznica et A. Kalousis, « Ontology-based meta-mining of knowledge discovery workflows », dans *Meta-Learning in Computational Intelligence*, 2011, p. 273–315.

[54] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank et M. Hall, « Multiclass alternating decision trees », dans *Proceedings of the 13th European Conference on Machine Learning*, 2002, p. 161–172.

[55] X. Hu, N. Sun, C. Zhang et T.-S. Chua, « Exploiting internal and external semantics for the clustering of short texts using world knowledge », dans *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, p. 919–928.

[56] A. Huang, « Similarity measures for text document clustering », dans *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, 2008, p. 49–56.

[57] D. Hull, « Improving text retrieval for the routing problem using latent semantic indexing », dans *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, p. 282–291.

[58] D. J. Ittner, D. D. Lewis et D. D. Ahn, « Text categorization of low quality images », dans *In Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995, p. 301–315.

[59] P. Jaccard, « Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines », dans *Bulletin de la Société Vaudoise des Sciences Naturelles*, t. 37, 1901, p. 241–272.

[60] N. Jankowski, W. Duch et K. Grabczewski, éds., *Meta-Learning in Computational Intelligence*. Springer, 2011, t. 358.

[61] T. Joachims, « A probabilistic analysis of the rocchio algorithm with tfidf for text categorization », dans *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, p. 143–151.

[62] T. Joachims, « Text categorization with suport vector machines : learning with many relevant features », dans *Proceedings of the 10th European Conference on Machine Learning*, 1998, p. 137–142.

[63] G. H. John et P. Langley, « Estimating continuous distributions in bayesian classifiers », dans *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1995, p. 338–345.

[64] K. S. Jones, S. Walker et S. E. Robertson, « A probabilistic model of information retrieval : development and comparative experiments », dans *Inf. Process. Manage.*, t. 36, nov. 2000, p. 779–808.

[65] K. S. Jones, « A statistical interpretation of term specificity and its application in retrieval », dans *Journal of documentation*, t. 28, 1972, p. 11–21.

[66] A. Kalousis, J. a. Gama et M. Hilario, « On data and algorithms : understanding inductive performance », dans *Mach. Learn.*, t. 54, mar. 2004, p. 275–312.

[67] A. Kalousis et M. Hilario, « Feature selection for meta-learning », dans *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2001, p. 222–233.

[68] A. Kalousis et M. Hilario, « Representational issues in meta-learning », dans *In Proceedings of the 20th International Conferente on Machine Learning*, 2003, p. 313–320.

[69] A. Kalousis et T. Theoharis, « Noemon : design, implementation and performance results of an intelligent assistant for classifier selection », dans *Intelligent Data Analysis*, t. 3, 1999, p. 319 –337.

[70] A. M. Kaptein, « Meta-classifier approach to reliable text classification », rap. tech., 2005.

[71] E. Keogh, S. Lonardi et C. A. Ratanamahatana, « Towards parameter-free data mining », dans *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, p. 206–215.

[72] A. M. Kibriya, E. Frank, B. Pfahringer et G. Holmes, « Multinomial naive bayes for text categorization revisited », dans *Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence*, 2004, p. 488–499.

[73] A. Kibriya, E. Frank, B. Pfahringer et G. Holmes, « Multinomial naive bayes for text categorization revisited », dans *AI 2004 : Advances in Artificial Intelligence*, t. 3339, 2005, p. 488–499.

[74] S.-B. Kim, K.-S. Han, H.-C. Rim et S.-H. Myaeng, « Some effective techniques for naive bayes text classification », dans *Knowledge and Data Engineering, IEEE Transactions on*, t. 18, 2006, p. 1457–1466.

[75] W. Konen, P. Koch, O. Flasch, T. Bartz-Beielstein, M. Friese et B. Naujoks, « Tuned data mining : a benchmark study on different tuners », dans *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, 2011, p. 1995–2002.

[76] C. Köpf, C. Taylor et J. Keller, « Meta-analysis : from data characterisation for meta-learning to meta-regression », dans *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP*, Citeseer, 2000.

[77] P. Kuba, P. Brazdil, C. Soares et A. Woznica, « Exploiting sampling and meta-learning for parameter setting for support vector machines », eng, dans *Proc. of Workshop Learning and Data Mining associated with Iberamia 2002, VIII Iberoamerican Conference on Artificial Intellignce*, 2002, p. 209–216.

[78] W. Lam et K.-Y. Lai, « A meta-learning approach for text categorization », dans *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, p. 303–309.

[79] M. Lan, S.-Y. Sung, H.-B. Low et C.-L. Tan, « A comparative study on term weighting schemes for text categorization », dans *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, t. 1, 2005, 546–551 vol. 1.

[80] M. Lan, C. Tan, J. Su et Y. Lu, « Supervised and traditional term weighting methods for automatic text categorization », dans *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, t. 31, 2009, p. 721–735.

[81] M. Lan, C. L. Tan, J. Su et Y. Lu, « Supervised and traditional term weighting methods for automatic text categorization », dans *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, t. 31, 2009, p. 721–735.

[82] R. Leite et P. Brazdil, « Improving progressive sampling via meta-learning on learning curves », dans *Machine Learning : ECML 2004*, t. 3201, 2004, p. 250–261.

[83] R. Leite et P. Brazdil, « Predicting relative performance of classifiers from samples », dans *Proceedings of the 22Nd International Conference on Machine Learning*, 2005, p. 497–503.

[84] R. Leite et P. Brazdil, « Active testing strategy to predict the best classification algorithm via sampling and metalearning », dans *Proceedings of the 2010 Conference on ECAI 2010 : 19th European Conference on Artificial Intelligence*, 2010, p. 309–314.

[85] R. Leite, P. Brazdil et J. Vanschoren, « Selecting classification algorithms with active testing », dans *Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition*, 2012, p. 117–131.

[86] E. Leopold et J. Kindermann, « Text categorization with support vector machines. how to represent texts in input space ? », dans *Mach. Learn.*, t. 46, mar. 2002, p. 423–444.

[87] V. Lertnattee, C. Leuviphan et al., « Using class frequency for improving centroid-based text classification », dans *ACEEE International Journal on Information Technology*, t. 2, 2012.

[88] V. Lertnattee et T. Theeramunkong, « Combining homogeneous classifiers for centroid-based text classification », dans *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, 2002, p. 1034–1039.

[89] V. Lertnattee et T. Theeramunkong, « Effect of term distributions on centroid-based text categorization », dans *Inf. Sci. Inf. Comput. Sci.*, t. 158, jan. 2004, p. 89–115.

[90] D. D. Lewis, « Naive (bayes) at forty : the independence assumption in information retrieval », dans *Proceedings of the 10th European Conference on Machine Learning*, 1998, p. 4–15.

[91] D. Lin et P. Pantel, « Dirt @sbt@discovery of inference rules from text », dans *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, p. 323–328.

[92] F. Lin et W. W. Cohen, « Semi-supervised classification of network data using very few labels », dans *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, 2010, p. 192–199.

[93] Y.-S. Lin, J.-Y. Jiang et S.-J. Lee, « A similarity measure for text classification and clustering », dans *IEEE Transactions on Knowledge and Data Engineering*, t. 99, 2013, p. 1.

[94] G. Linden, B. Smith et J. York, « Amazon.com recommendations : item-to-item collaborative filtering », dans *IEEE Internet Computing*, t. 7, jan. 2003, p. 76–80.

[95] C. D. Manning, P. Raghavan et H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[96] S. Martin, J. Liermann et H. Ney, « Algorithms for bigram and trigram word clustering », dans *Speech Commun.*, t. 24, avr. 1998, p. 19–37.

[97] J. Martineau, T. Finin, A. Joshi et S. Patel, « Improving binary classification on text problems using differential word features », dans *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, p. 2019–2024.

[98] A. McCallum, K. Nigam et al., « A comparison of event models for naive bayes text classification », dans *AAAI-98 workshop on learning for text categorization*, 1998, p. 41–48.

[99] C. J. Merz, « Dynamical selection of learning algorithms », dans *Learning from Data*, 1996, p. 281–290.

[100] D. Michie, D. J. Spiegelhalter, C. C. Taylor et J. Campbell, éds., *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[101] G. A. Miller, « Wordnet : a lexical database for english », dans *Commun. ACM*, t. 38, nov. 1995, p. 39–41.

[102] P. B. C. de Miranda, R. B. C. Prudêncio, A. C. P. L. F. de Carvalho et C. Soares, « Combining a multi-objective optimization approach with meta-learning for svm parameter selection », dans *SMC*, 2012, p. 2909–2914.

[103] T. M. Mitchell, *Machine Learning*, 1re éd. McGraw-Hill, Inc., 1997.

[104] M. M. Molina, J. M. Luna, C. Romero et S. Ventura, « Meta-learning approach for automatic parameter tuning : a case study with educational datasets », dans *Proceedings of the 5th International Conference on Educational Data Mining, EDM 2012*, 2012, p. 180–183.

[105] S. Moran, Y. He et K. Liu, « Choosing the best bayesian classifier : an empirical study », dans *IAENG International Journal of Computer Science*, t. 36, 2009, p. 322–331.

[106] T. Mori, « Information gain ratio as term weight : the case of summarization of ir results », dans *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, 2002, p. 1–7.

[107] R. Moschitti et R. Basili, « Complex linguistic features for text classification : a comprehensive study », dans *Proceedings of the 26th European Conference on Information Retrieval (ECIR*, 2004, p. 181–196.

[108] D. Nakache et E. Metais, « Evaluation : nouvelle approche avec juges », dans *INFORSID'05 XXIII e congrès, Grenoble*, 2005, p. 555–570.

[109] P. Nakov et M. A. Hearst, « Solving Relational Similarity Problems Using the Web as a Corpus », dans *Proceedings of ACL-08 : HLT*, juin 2008, p. 452–460.

[110] P. Nguyen, J. W. 0017, M. Hilario et A. Kalousis, « Learning heterogeneous similarity measures for hybrid-recommendations in meta-mining », dans *CoRR*, t. abs/1210.1317, 2012.

[111] T. T. Nguyen, K. Chang et S. C. Hui, « Word cloud model for text categorization », dans *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, IEEE, 2011, p. 487–496.

[112] R. Pavón, F. Díaz, R. Laza et V. Luzón, « Automatic parameter tuning with a bayesian case-based reasoning system. a case of study », dans *Expert Syst. Appl.*, t. 36, mar. 2009, p. 3407–3420.

[113] M. Pechenizkiy, « Data mining strategy selection via empirical and constructive induction. », dans *Databases and Applications*, 2005, p. 59–64.

[114] Y. Peng, P. A. Flach, C. Soares et P. Brazdil, « Improved dataset characterisation for meta-learning », dans *Proceedings of the 5th International Conference on Discovery Science*, 2002, p. 141–152.

[115] B. Pfahringer, H. Bensusan et C. G. Giraud-Carrier, « Meta-learning by landmarking various learning algorithms », dans *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, p. 743–750.

[116] X.-H. Phan, L.-M. Nguyen et S. Horiguchi, « Learning to classify short and sparse text & web with hidden topics from large-scale data collections », dans *Proceedings of the 17th International Conference on World Wide Web*, 2008, p. 91–100.

[117] J. C. Platt, « Advances in kernel methods », dans, 1999, chap. Fast training of support vector machines using sequential minimal optimization, p. 185–208.

[118] R. B. C. Prudêncio et T. B. Ludermir, « Selective generation of training examples in active meta-learning », dans *Int. J. Hybrid Intell. Syst.*, t. 5, avr. 2008, p. 59–70.

[119] R. B. C. Prudêncio et T. B. Ludermir, « Combining uncertainty sampling methods for supporting the generation of meta-examples », dans *Inf. Sci.*, t. 196, août 2012, p. 1–14.

[120] R. B. C. Prudencio, T. B. Ludermir et F. de A.T. de Carvalho, « A modal symbolic classifier for selecting time series models », dans *PATTERN RECOGN. LETTERS*, t. 25, 2004, p. 911–921.

[121] R. B. C. Prudêncio et T. B. Ludermir, « Combining uncertainty sampling methods for active meta-learning », dans *ISDA*, 2009, p. 220–225.

[122] J. R. Quinlan, *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[123] M. Reif, « A comprehensive dataset for evaluating approaches of various meta-learning tasks », dans *Proceedings of the First International Conference on Pattern Recognition Applications and Methods*, fév. 2012.

[124] M. Reif, F. Shafait et A. Dengel, « Prediction of classifier training time including parameter optimization », dans *Proceedings of the 34th Annual German Conference on Advances in Artificial Intelligence*, 2011, p. 260–271.

[125] M. Reif, F. Shafait et A. Dengel, « Meta-learning for evolutionary parameter optimization of classifiers », dans *Mach. Learn.*, t. 87, juin 2012, p. 357–380.

[126] J. D. M. Rennie, L. Shih, J. Teevan et D. R. Karger, « Tackling the poor assumptions of naive bayes text classifiers », dans *In Proceedings of the Twentieth International Conference on Machine Learning*, 2003, p. 616–623.

[127] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom et J. Riedl, « Grouplens : an open architecture for collaborative filtering of netnews », dans *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 1994, p. 175–186.

[128] J. R. Rice, « The algorithm selection problem », t. 15, 1976, p. 65–118.

[129] S. E. Robertson et K. S. Jones, « Relevance weighting of search terms », dans *Journal of the American Society for Information science*, t. 27, 1976, p. 129–146.

[130] S. Robertson, « Understanding inverse document frequency : on theoretical arguments for idf », dans *Journal of Documentation*, t. 60, 2004, p. 2004.

[131] J. J. Rocchio, « Relevance feedback in information retrieval », dans *The Smart retrieval system - experiments in automatic document processing*, 1971, p. 313–323.

[132] D. J. Rogers et T. T. Tanimoto, « A Computer Program for Classifying Plants », dans *Science*, t. 132, oct. 1960, p. 1115–1118.

[133] M. E. Ruiz et P. Srinivasan, « Hierarchical text categorization using neural networks », dans *Inf. Retr.*, t. 5, jan. 2002, p. 87–118.

[134] G. Salton, *The SMART Retrieval System. Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971.

[135] G. Salton, A. Wong et C. S. Yang, « A vector space model for automatic indexing », dans *Commun. ACM*, t. 18, nov. 1975, p. 613–620.

[136] G. Salton, *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[137] G. Salton et C. Buckley, « Term-weighting approaches in automatic text retrieval », dans *Information processing & management*, t. 24, 1988, p. 513–523.

[138] G. Salton et M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[139] S. Salzberg, « A nearest hyperrectangle learning method », dans *Mach. Learn.*, t. 6, mai 1991, p. 251–276.

[140] C. Schaffer, « A conservation law for generalization performance. », dans *ICML*, t. 94, 1994, p. 259–265.

[141] K.-M. Schneider, « Techniques for improving the performance of naive bayes for text classification », dans *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, 2005, p. 682–693.

[142] F. Sebastiani, « Machine learning in automated text categorization », dans *ACM Comput. Surv.*, t. 34, mar. 2002, p. 1–47.

[143] S. Segrera, J. Pinho et M. Moreno, « Information-theoretic measures for meta-learning », dans *Hybrid Artificial Intelligence Systems*, t. 5271, 2008, p. 458–465.

[144] F. Serban, J. Vanschoren, J.-U. Kietz et A. Bernstein, « A survey of intelligent assistants for data analysis », dans *ACM Comput. Surv.*, t. 45, juil. 2013, 31 :1–31 :35.

[145] B. Settles, « Active learning literature survey », dans *University of Wisconsin, Madison*, t. 52, 2010, p. 55–66.

[146] S. Shankar et G. Karypis, « Weight adjustment schemes for a centroid based classifier », Department of Computer Science et Engineering - University of Minnesota, rap. tech., 2000.

[147] C. E. Shannon, « A mathematical theory of communication », dans *ACM SIGMOBILE Mobile Computing and Communications Review*, t. 5, 2001, p. 3–55.

[148] J. W. Shavlik, R. J. Mooney et G. G. Towell, « Symbolic and neural learning algorithms : an experimental comparison », dans *Machine learning*, t. 6, 1991, p. 111–143.

[149] Z. Shevked et L. Dakovski, « Learning and classification with prime implicants applied to medical data diagnosis », dans *Proceedings of the 2007 International Conference on Computer Systems and Technologies*, 2007, 103 :1–103 :5.

[150] A. F. Smeaton, « Using nlp or nlp resources for information retrieval tasks », dans *Natural Language Information Retrieval*, 1997, p. 99–111.

[151] K. A. Smith-Miles, « Cross-disciplinary perspectives on meta-learning for algorithm selection », dans *ACM Comput. Surv.*, t. 41, jan. 2009, 6 :1–6 :25.

[152] C. Soares et P. B. Brazdil, « Selecting parameters of svm using meta-learning and kernel matrix-based meta-features », dans *Proceedings of the 2006 ACM Symposium on Applied Computing*, 2006, p. 564–568.

[153] C. Soares, P. B. Brazdil et P. Kuba, « A meta-learning method to select the kernel width in support vector regression », dans *Machine Learning*, t. 54, 2004, p. 195–209.

[154] C. Soares et P. Brazdil, « Zoomed ranking : selection of classification algorithms based on relevant performance information », dans *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 2000, p. 126–135.

[155] S. Y. Sohn, « Meta analysis of classification algorithms for pattern recognition », dans *IEEE Trans. Pattern Anal. Mach. Intell.*, t. 21, nov. 1999, p. 1137–1144.

[156] P. Soucy et G. W. Mineau, « Beyond tfidf weighting for text categorization in the vector space model », dans *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005, p. 1130–1135.

[157] M. C. P. de Souto, R. B. C. Prudêncio, R. G. Soares, D. S. de Araujo, I. G. Costa, T. B. Ludermir et A. Schliep, « Ranking and selecting clustering algorithms using a meta-learning approach », dans *Neural Networks*, IEEE, 2008, p. 3729–3735.

[158] M. Spiliopoulou, A. Kalousis, L. C. Faulstich et T. Theoharis, « Noemon : an intelligent assistant for classifier selection », dans *FGML98*, t. 98, 1998, p. 90–97.

[159] B. Srivastava et A. Mediratta, « Domain-dependent parameter selection of search-based algorithms compatible with user performance criteria », dans *AAAI*, 2005, p. 1386–1391.

[160] J. Su, H. Zhang, C. X. Ling et S. Matwin, « Discriminative parameter learning for bayesian networks », dans *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, p. 1016–1023.

[161] A. Sun, « Short text classification using very few words », dans *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, p. 1145–1146.

[162] Q. Sun et B. Pfahringer, « Pairwise meta-rules for better meta-learning-based algorithm ranking », English, dans *Machine Learning*, t. 93, 2013, p. 141–161.

[163] V. Tam, A. Santoso et R. Setiono, « A comparative study of centroid-based, neighborhood-based and statistical approaches for effective document categorization », dans *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02)*, t. 4, 2002, p. 235–238.

[164] S. Tan, « Large margin dragpushing strategy for centroid text categorization », dans *Expert Systems with Applications*, t. 33, juil. 2007, p. 215–220.

[165] S. Tan, « An improved centroid classifier for text categorization », dans *Expert Systems with Applications*, t. 35, 2008, p. 279–285.

[166] D. M. Tax et R. P. Duin, « Characterizing one-class datasets », dans *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2005, p. 21–26.

[167] T. Theeramunkong et V. Lertnattee, « Improving centroid-based text classification using term-distribution-based weighting system and clustering », dans *Proceedings of ISCIT-01, 2nd International Symposium on Communication and Information Technology, Cheingmai, Thailand*, 2001, p. 1167–1182.

[168] L. Todorovski, P. Brazdil et C. Soares, « Report on the experiments with feature selection in meta-level learning », dans *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP : forum for practical problem presentation and prospective solutions*, Citeseer, 2000.

[169] P. D. Turney, « Similarity of semantic relations », dans *Comput. Linguist.*, t. 32, sept. 2006, p. 379–416.

[170] P. D. Turney et M. L. Littman, « Measuring praise and criticism : inference of semantic orientation from association », dans *ACM Trans. Inf. Syst.*, t. 21, oct. 2003, p. 315–346.

[171] P. D. Turney et P. Pantel, « From frequency to meaning : vector space models of semantics », dans *J. Artif. Int. Res.*, t. 37, jan. 2010, p. 141–188.

[172] R. Vilalta et Y. Drissi, « A perspective view and survey of meta-learning », dans *Artificial Intelligence Review*, t. 18, 2002, p. 77–95.

[173] R. Vilalta, C. G. Giraud-Carrier, P. Brazdil et C. Soares, « Using meta-learning to support data mining. », dans *IJCSA*, t. 1, 2004, p. 31–45.

[174] E. M. Voorhees, « Using wordnet to disambiguate word senses for text retrieval », dans *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993, p. 171–180.

[175] M. Wajeed et T. Adilakshmi, « Different similarity measures for text classification using knn », dans *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on*, 2011, p. 41–45.

[176] C Van der Walt et E. Barnard, « Data characteristics that determine classifier performance. », dans *17th Annual Symposium of the Pattern Recognition Association of South Africa*, 2006.

[177] S. M. Weiss et I. Kapouleas, « An empirical comparison of pattern recognition, neural nets, and machine learning classification methods », dans *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, 1989, p. 781–787.

[178] S. M. Weiss et C. A. Kulikowski, *Computer Systems That Learn : Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann Publishers Inc., 1991.

[179] E. B. Wilson, « Probable inference, the law of succession, and statistical inference », dans *Journal of the American Statistical Association*, t. 22, 1927, p. 209–212.

[180] I. H. Witten et E. Frank, *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers Inc., 2000.

[181] D. H. Wolpert, « The supervised learning no-free-lunch theorems », dans *In Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, 2001, p. 25–42.

[182] D. H. Wolpert et W. G. Macready, « No free lunch theorems for search », Technical Report SFI-TR-95-02-010, Santa Fe Institute, rap. tech., 1995.

[183] D. H. Wolpert et W. G. Macready, « No free lunch theorems for optimization », dans *Evolutionary Computation, IEEE Transactions on*, t. 1, 1997, p. 67–82.

[184] H. Wu et G. Salton, « A comparison of search term weighting : term relevance vs. inverse document frequency », dans *Proceedings of the 4th Annual International ACM SIGIR Conference on Information Storage and Retrieval : Theoretical Issues in Information Retrieval*, 1981, p. 30–39.

[185] Y. Wu et D. W. Oard, « Bilingual topic aspect classification with a few training examples », dans *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008, p. 203–210.

[186] J. Xu et W. B. Croft, « Corpus-based stemming using cooccurrence of word variants », dans *ACM Transactions on Information Systems (TOIS)*, t. 16, jan. 1998, p. 61–81.

[187] Y. Yang et J. O. Pedersen, « A comparative study on feature selection in text categorization », dans *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, p. 412–420.

[188] S. Zelikovitz, W. W. Cohen et H. Hirsh, « Extending whirl with background knowledge for improved text classification », dans *Inf. Retr.*, t. 10, jan. 2007, p. 35–67.

[189] H.-J. Zeng, X.-H. Wang, Z. Chen, H. Lu et W.-Y. Ma, « Cbc : clustering based text classification requiring minimal labeled data », dans *Proceedings of the Third IEEE International Conference on Data Mining*, IEEE, 2003, p. 443–450.

[190] X. Zhang, T. Wang, X. Liang, F. Ao et Y. Li, « A class-based feature weighting method for text classification », dans *Journal of Computational Information Systems*, t. 8, 2012, p. 965–972.

[191] X. Zhu, « Semi-supervised learning literature survey », dans *world*, t. 10, 2005, p. 10.

[192] J. Zobel et A. Moffat, « Exploring the similarity space », dans *ACM SIGIR Forum*, ACM, t. 32, 1998, p. 18–34.