REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE
N° :……..

DOMAINE : MATHEMATIQUES ET INFORMATIQUE
FILIERE : INFORMATIQUE
OPTION : RESEAUX ET TECHNOLOGIE DE L'INFORMATION ET DE LA COMMUNICATION

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: Gherabi Charaf Eddine

Intitulé

Détection des spams se basant sur les techniques de classification

Soutenu devant le jury composé de :

Mlle. Saoudi Lalia	Université de M'sila	Président
Mr. Chikouche Noureddine	Université de M'sila	Rapporteur
Mr. Kamel Mohamed	Université de M'sila	Examinateur

Année universitaire : 2017 /2018

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION GENERALE

Le courrier électronique (ou courriel, email) est un des services les plus utilisés sur internet, il est sans doute la technique qui a changé nos habitudes à une grande échelle. La croissance de l'Internet est reliés directement à l'importance du courriel, car plusieurs sites web lui sont maintenant consacrés, et presque tous les gens qui ont accès à internet ont au moins une adresse de courrier électronique qu'ils vérifient quotidiennement, ce qui explique les milliards des courriels qui s'envoient et sont reçus chaque jour.

Aujourd'hui, le courriel rend vraiment service aux usagers, c'est un moyen rapide et économique pour échanger des informations. Si nous comparons le courrier électronique aux autres moyens de communication, (par écrit, téléphone), nous nous apercevons que les avantages des courriels surpassent ses inconvénients. Sa force réside dans le médium du transport des messages, la rapidité avec laquelle circulent les courriels, l'économie, la disponibilité en tout temps indépendamment du décalage horaire et à la possibilité de les envoyer à plusieurs personnes en même temps. La nature informatique de ces courriels offre des avantages incomparables, dont l'envoi des documents électroniques par attachement, l'archivage des messages est beaucoup plus facile à effectuer qu'avec les communications écrites ou par téléphone, ainsi que, le courrier électronique permet d'effectuer un traitement rapide, efficace et automatique sur les messages comme la recherche par mots clés, le tri automatique par sujet.

Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de courriers électroniques indésirables ou non sollicités appelés aussi spam. En effet, le spam est rapidement devenu un problème majeur sur Internet.

Le spam est un phénomène mondial et massif. Selon la CNIL (La Commission Nationale de l'Informatique et des Libertés), le spam est défini de la manière suivante : « Le "spamming" ou "spam" est l'envoi massif de courriers électroniques non sollicités, à des personnes avec lesquelles l'expéditeur n'a jamais eu de contact et dont il a capté l'adresse électronique de façon irrégulière. », Il existe de nombreuses techniques contre le spam qui peuvent être divisées en deux groupes [1]. Le premier contient les solutions basées sur l'en-tête du message électronique telles que les listes noires et les listes blanches. Le deuxième groupe de solutions contient celles qui sont basées sur le contenu textuel du message telles que le filtrage basé sur l'apprentissage automatique.

Il existe de nombreux travaux qui traitent le problème de filtrage de spam en utilisant des méthodes d'apprentissage automatique [2]. Le filtrage de spam basé sur le contenu textuel des messages peut être considéré comme un exemple de classification de textes qui consiste en l'attribution de documents textuels à un ensemble de classes prédéfinies. Le but d'un système de classification est d'effectuer la tâche de classification et de le faire avec un degré raisonnable d'exactitude. Il existe aujourd'hui une liste plutôt longue de classifieurs développés autour des algorithmes.

En va faire dans le cadre de ce travaille une étude comparative se basant sur trois algorithmes d'apprentissage : Machine à Vecteur de support, Naïve bayes et K Plus Proche Voisin. L'objectif principal est de sélectionner le meilleur algorithme pour classifier le courrier électronique reçu.

Nous avons décomposé notre mémoire en quatre chapitres. Le premier chapitre vise à présenter la définition de spam, à travers ses objectifs, ses contenus et ses impacts et aussi les différentes techniques utilisées pour détecter ce type de courriels, le deuxième chapitre consiste à présenter la classification des textes puisque le filtrage de spam est considéré comme une tâche de classification de textes, le troisième chapitre expose quelques travaux publiés sur le filtrage de spam, le quatrième chapitre décrit l'architecture du système, les outils utilisés pour l'implémentation ainsi que résultats expérimentaux réalisés qui sont présentés avec des figures illustratives, afin de faciliter la compréhension et la comparaison de ces résultats pour choisir le meilleur classifieur.

CHAPITRE 1

DETECTION ET FILTRAGE DES SPAMS

1.1. Introduction

Le spam est un grand problème pour les internautes. Les augmentations récentes du taux de spam ont causé une grande inquiétude parmi la communauté Internet. De nombreuses solutions avaient été suggérées pour résoudre le problème.

Dans ce chapitre, nous présentons tout d'abord les débuts du spam, ses objectifs, ses contenus, ses impacts et les différentes techniques utilisées pour détecter ce type de courriels.

1.2. Naissance et débuts du spam

1.2.1. Origine du mot spam

En 1937 La société Hormel Foods[1] organise un concours pour trouver un nouveau nom pour leur jambon épicé, Ce nom doit être aussi caractéristique que le goût du produit « **Sp**iced H**am** » et qui propose « Spam » pour ce produit, fut donc la marque retenue.

Cette viande précuite en boîte souvent synonyme de mauvaise nourriture a été largement utilisée par l'intendance des forces armées américaines pour la nourriture des soldats pendant la Seconde Guerre mondiale et sera introduite dans diverses régions du monde à cette occasion. [3]

1.2.2. Définition du spam

Le spam est un message électronique non sollicité, envoyé massivement à un grand nombre de destinataires, à des fins publicitaires ou malveillantes. [3]

Le terme spam est aussi utilisé pour désigner le même type de message transmis par d'autres moyens de communication électroniques tels que les messageries instantanées, les blogs, les forums, et plus récemment, des réseaux de téléphonie mobile, via les SMS ou MMS. Même si le moyen de communication est différent, les techniques d'envoi et de détection restent relativement similaires.

Le premier spam (Figure 1.1**)** date du 3 mai 1978. Ce jour là, sur le réseau ARPANET[2] , Gary Thuerk, commercial de la société informatique DEC[3], invitait par e-mail 393 personnes à découvrir sa nouvelle machine, le 2020.

[1] Hormel Foods : fabricant de viande en conserve
[2] ARPANET : est le premier réseau à transfert de paquets développé aux États-Unis
[3] DEC : Digital Equipment Corporation

Le message se présentait ainsi :

Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT
Date: 1 May 1978 1233-EDT
From: THUERK at DEC-MARLBORO
Subject: ADRIAN@SRI-KL

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 – 2 PM
HYATT HOUSE (NEAR THE L.A. AIRPORT)
LOS ANGELES, CA

THURSDAY, MAY 11, 1978 – 2 PM
DUNFEY'S ROYAL COACH
SAN MATEO, CA
(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.

Figure 1.1 Le premier spam

Ce message indésirable n'était hélas que le premier d'une longue série. Le spam était né. [4]

1.3. Objectifs et statistiques sur les spam

Au départ, le spam visait principalement des objectifs publicitaires. Aujourd'hui, il s'est considérablement développé, diversifié et complexifié, pour atteindre de plus en plus souvent des objectifs malveillants. En effet, Le spam s'est non seulement développé en termes de volume, mais également en termes de contenu (voir figure 1.2). Aujourd'hui, les objectifs des spam sont très variés en voici une liste non exhaustive:

- ***Hameçonnage*** **(ou phishing) :** L'objectif est de réussir à se faire passer pour un organisme connu par l'utilisateur, dans le but de lui voler des informations à caractère confidentiel. Par exemple, on reçoit un mail provenant "apparemment" de notre banque, ou d'un autre site où l'on dispose d'informations personnelles. dans ce mail, il est demandé de cliquer sur un lien (pour des motifs divers :

réactualisation, etc.), après avoir cliqué sur ce lien, une page web s'affiche... sur laquelle il est demandé de rentrer ses coordonnées bancaires ou toute autre information personnelle. Parmi les sites Top les plus contrefaits pour les attaques de phishing, on retrouve eBay, Paypal et Bank of America. [5]

- ***Publicité*** **:** L'objectif est de vanter les mérites d'un produit quelconque. Il s'agit par exemple de produits pharmaceutiques, de produits de luxe, de logiciels divers et variés, de jeux d'argent. Ils peuvent également soutenir-agate idées politiques, culturelles ou religieuses et / ou organisations.
- ***Scam*** **:** Il s'agit d'une attaque basé sur la naïveté des destinataires dans le but de leur soutirer de l'argent. L'exemple le plus courant est le scam nigérien: un dignitaire d'un pays d'afrique vous demande de servir d'intermédiaire pour une transaction financière importante, en vous promettant un bon pourcentage de la somme. Pour amorcer la transaction, il vous faut donner de l'argent. [6]
- ***Canular*** **:** L'objectif est de faire circuler une information semblant très sensible, souvent avec un caractère d'urgence : fausse alerte de virus, fausse alerte de contamination potentielle, chaine de solidarité….. Par exemple : « *un nouveau virus très dangereux se propage, il faut faire circuler l'information* » ; « *des sous-vêtements sont infectés par une dangereuse bactérie* ».
- ***Malware*** **:** Est un logiciel conçu pour infiltrer ou endommager un système informatique. Il est communément pris pour contenir des virus informatiques, vers, chevaux de Troie, spywares et adwares. Ce type de logiciel est souvent envoyé en tant que non suspect d'une pièce jointe. Lorsque l'utilisateur ouvre le fichier, le logiciel malveillant s'installe. L'interdépendance entre les spams et les logiciels malveillants a évolué Spam logiciels malveillants propagation des e-mails, les logiciels malveillants est utilisé pour infecter un hôte de sorte que l'hôte peut être contrôlé à distance et utilisé pour l'envoi de plus de spams. Ces hôtes infectés sont désignées comme des «ordinateurs zombies». Beaucoup de gens croient que la plupart des spams sont envoyés par des botnets, qui constituent un réseau de PC zombies. [7]

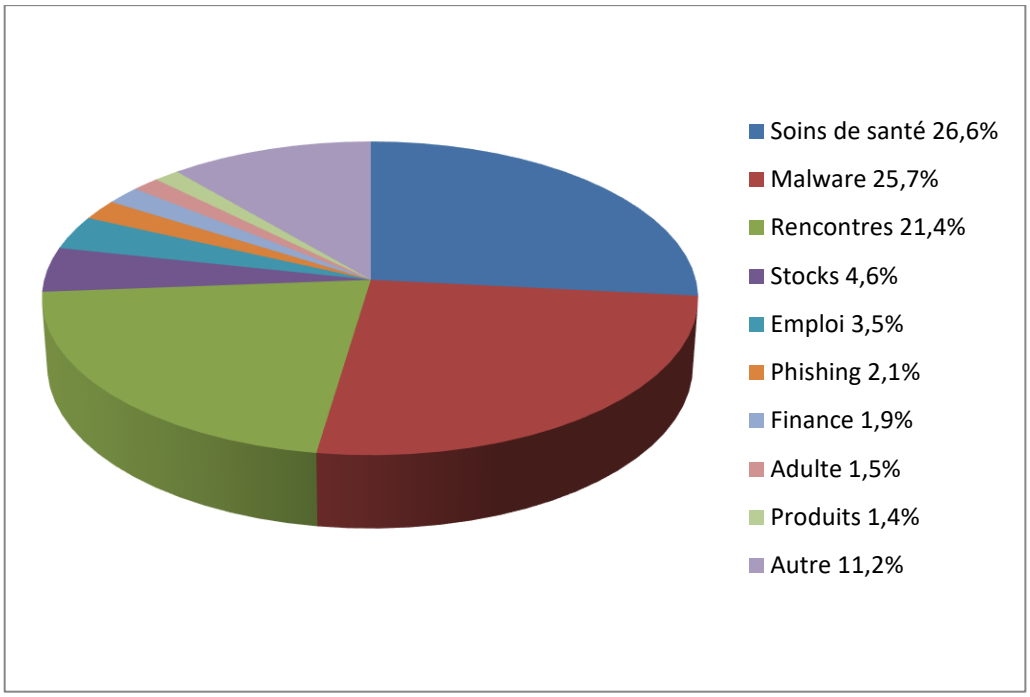


Figure 1.2 Répartition des spam par contenu [8]

On a quelques statistiques sur le taux global de spam entre les années 2012 et 2017 présenté dans la figure 1.4. Dans la dernière période il a été constaté que le spam représentait 55% de tous les messages électroniques, comme au cours de l'année précédente.

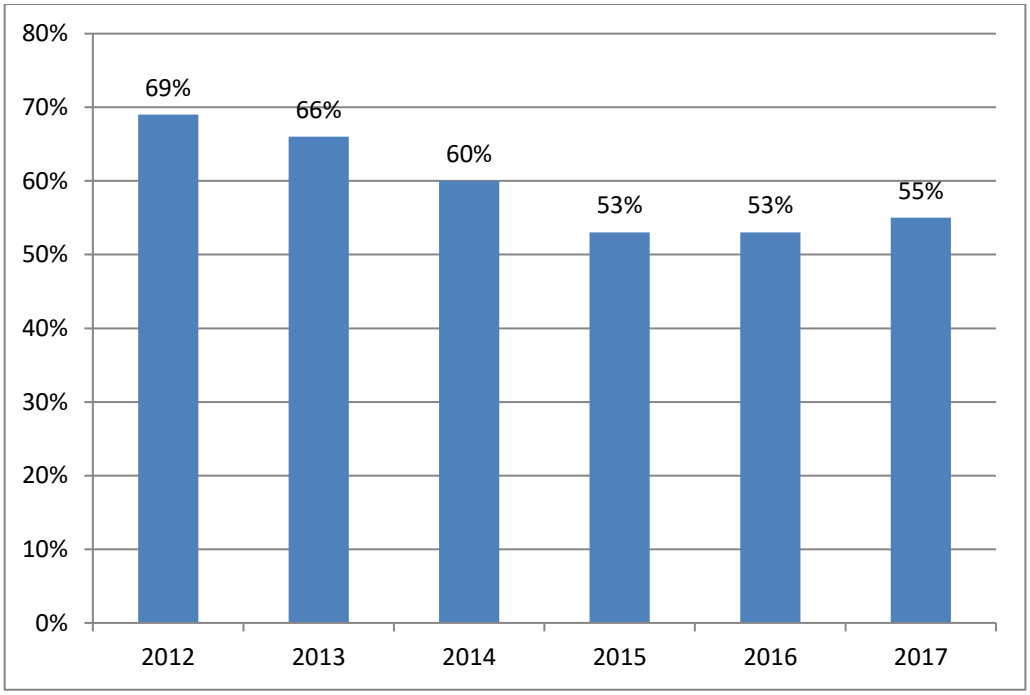


Figure 1.3 développement de spam en termes de volume [8]

1.4. Impactes du spam sur les utilisateurs et les fournisseurs

Dans cette section, nous présentons les effets du spam, au niveau des utilisateurs, entreprises et FAI. [9]

1.4.1. Perte de temps :

- Encombrement anormal des boîtes aux lettres.
- Suppression des courriels indésirables.
- Configuration et maintenance des filtres.
- Consultation des courriels rejetés pour y détecter les bons à cause du risque de passer à côté d'emails importants mal catalogués par les outils de détection anti-spam.

1.4.2. Perte de bande passante et d'espace disque :

- Spécialement pour les utilisateurs de modems.
- Les pièces jointes des virus et spam peuvent être grands.

1.4.3. Pertes financières non négligeables aux niveaux des entreprises et FAI :

- une augmentation des coûts de gestion opérationnelle et support lié à la gestion anti spam.
- perte de productivité des salariés,

Selon une étude, le spam aurait coûté environ 712 $ par employé et par an aux entreprises. À ce chiffre, il faut rajouter 113 à 183 $ par employé et par an pour la gestion des emails en quarantaine.

1.5. Techniques de filtrage du spam

Plusieurs techniques de lutte contre le spam sont possibles et peuvent être cumulées : analyse statistique (filtre bayésien), filtrage par mots clés, listes blanches, listes noires. Ces techniques de lutte doivent s'adapter en permanence car de nouveaux types de spam réussissent à les contourner.

Deux solutions de détection de spam sont envisageables : la détection au niveau du serveur mail FAI et la détection au niveau de l'utilisateur final.

Ces outils peuvent être divisés en deux groupes : le filtrage d'enveloppe, et le filtrage de contenu.

1.5.1. Filtrage d'enveloppe

Ce type de filtrage s'applique uniquement sur l'en-tête[4] du message, qui contient souvent assez d'informations pour pouvoir distinguer un spam. Cette technique appliquée au niveau du serveur FAI présente l'avantage de pouvoir bloquer les courriels avant même que leur corps ne soit envoyé, ce qui diminue grandement le trafic sur la passerelle SMTP.

Dans cette catégorie, nous trouvons les techniques suivantes :

1.5.1.1. Filtrage par listes noires :

Ces listes consistent à pré-déclarer une liste de « mauvais expéditeurs », (adresses emails, noms des domaines, pays, adresse IP), desquelles le destinataire refuse de recevoir des emails. Ces listes peuvent être :

- créées par l'administrateur ou l'utilisateur.
- téléchargées via le web (cela nécessite une mise à jour très régulière pour un filtrage optimisé)
- consultées en temps réel sur le web (RBL, Real Time Blackhole List).

Pour contourner ces listes les spammeurs, changent très fréquemment leurs adresses d'expédition (email, ou IP). [10]

1.5.1.2. Filtrage par listes blanches :

Ces listes consistent à pré-déclarer une liste de (adresses emails, noms des domaines, adresse IP) sûres desquels le destinataire accepte de recevoir des emails. Par défaut très peu d'hôtes sont considérés comme sûrs car leurs adresses pourraient être usurpées par les spammeurs. Tout comme la liste noire, la liste blanche a également besoin d'une mise à niveau continue et de rafraîchissement. [11]

1.5.1.3. Filtrage par liste grise :

La liste grise est un mixte entre la liste blanche et la liste noire. Ce qui se produit est qu'à chaque fois qu'une boîte aux lettres donnée reçoit un email d'un contact inconnu, cet email est suspendu avec un message de réponse automatique contenant un lien permettant de valider

[4] Elle constitue les informations de base de ce dernier : expéditeur, destinataire, date d'envoi, serveur source, Objet.

l'envoi. Ceci à pour but de détecter les robots, les spammeurs ne se rendront pas compte qu'ils doivent émettre une validation afin que le message soit accepté.

Dans le cas d'un réel email attendu et que l'expéditeur n'est pas énumérée dans l'une ou l'autre des listes noire et blanche, alors il sera positionné en liste grise. Si l'expéditeur satisfait la demande de confirmation (souvent un lien Web à cliquer), il obtiendra alors le passage à liste blanche et ses messages vous seront acheminés. C'est en fait l'ouverture dynamique de la liste blanche.

Par exemple : la figure suivante présente une réponse de cette technique.

Subject: Re: Hi There!

Greetings,

You just sent an email to my Spam-free email service. Because this is the first time you have sent to this email ac count, please confirm yourself so you'll be recognized when you send to me in the future. It's easy. To prove

your message comes from a human and not a computer, click on the link below:

http://[Some Web Link]

Attached is your original message that is in my pending folder, waiting for your quick authentication.

Figure 1.4 Exemple sur une réponse de technique de la liste grise [10]

1.5.1.4. Filtrage par vérification du domaine :

Les destinataires sont configurés de sorte qu'ils n'acceptent que les messages provenant de domaines spécifiques. Les e-mails dont les domaines ne sont pas mentionnés ne seront pas reçus. De cette façon, beaucoup de spam est bloqué. [12]

1.5.2. Filtrage du contenu

Ce type de filtrage se fait au niveau de l'utilisateur où son contenu est analysé pour détecter les spam qui ont réussi à passer à travers le filtre d'enveloppe.

Dans cette catégorie nous trouvons les techniques suivantes :

1.5.2.1. Filtrage par mots clés :

L'administrateur doit indiquer la liste des mots clés à détecter afin de déterminer qu'un mail est un Spam. Par exemple, tous les emails qui contiennent les mots : viagra, argent, money, drogue seront détectés comme Spam.

Ce filtre se base sur les mots clé inclus dans les mails. L'analyse est très rapide, mais peux efficace. Car cela demande un suivi manuel et les Spammeurs font varier les mots clé afin d'éviter ce filtre. Par exemple, on retrouve *M.O.N.E.Y.* ou encore *m*o*n*e*y*. [13]

1.5.2.2. Filtrage par caractères :

Il s'agit de bloquer les emails qui contiennent certains caractères ou police de caractère, ou certaines langues utilisées dans ces emails.

1.5.2.3. Filtrage d'image :

Il s'agit d'analyser les images obtenues dans les messages au niveau des propriétés du fichier image (format, taille du fichier, taille d'image) que du contenu de l'image (couleurs, test de pixels,…).

1.5.2.4. Filtrage d'URL :

Ceci consiste à vérifier les liens hypertextes inclus dans les messages auprès d'une base de données de « mauvais URL » préenregistrés, ou via la consultation en temps réel des listes noires disponibles sur le web. Des tentatives de masquage du lien hypertexte sont des fois utilisées par des spammeurs pour empêcher l'analyse par le filtrage d'URL.

1.5.2.5. Filtres bayésiens :

L'approche d'apprentissage automatique le plus connu dans le filtrage des spams est le classificateurs Bayes naïfs, classificateur Naïve Bayes est un classificateur probabiliste. En bref, il calcule et utilise la probabilité de certains mots / expressions apparaissant dans les exemples les plus connus (messages) afin de classer de nouveaux exemples (messages). Naïve Bayes a été montré pour être très bien réussi à catégoriser les documents texte. Filtres bayésiens (méthode statistique) Filtres travaillé en analysant les mots du message à l'intérieur d'un e-mail pour calculer la probabilité que le message est un spam ou non. Le calcul basé sur

des mots qui déterminent que le message est un spam et les mots qui déterminent que le message n'est pas du spam.

1.5.2.6. Machine à Vecteurs de Support (SVM) :

Machine à Vecteurs de Support (SVM) ont eu du succès dans le classement des documents texte. SVM a donné lieu à une recherche importante dans les appliquer à filtrage de spam. SVM sont des méthodes à noyaux dont l'idée centrale est d'intégrer les données représentant les documents texte dans un espace vectoriel. SVM tenter de construire une séparation linéaire entre deux classes dans cet espace vectoriel.

Une machine à vecteurs de support est un classifieur linéaire binaire à marge maximale. Il peut être interprété comme trouver un hyperplan dans un espace de caractéristiques linéairement séparables qui sépare les deux classes avec une marge maximum. Les instances les plus proches de l'hyperplan sont connues comme les « vecteurs de support » car ils soutiennent l'hyperplan des deux côtés de la marge.

SVM a été rapporté significative des performances sur le problème de la catégorisation de textes avec de nombreuses fonctionnalités pertinentes. SVM a également été appliquée au filtrage anti-spam. [7]

1.6. Conclusion

Dans ce chapitre en à présente de définitions de spam, ses objectifs et impacts ainsi les différents approches a battant de spam en peu divise en deux catégories : Le premier contient les solutions basées sur l’en-tête du message électronique telles que les listes noires, blanches et grises. Le deuxième groupe de solutions contient celles qui sont basées sur le contenu textuel du message telles que le filtrage basé sur l’apprentissage automatique.

CHAPITRE 2

LA CLASSIFICATION DES TEXTES

2.1. Introduction

La Classification (ou Catégorisation) de textes est aujourd'hui un domaine de recherche bien établi et très actif. Les travaux portent depuis une quinzaine d'année sur les systèmes avec apprentissage des classes à partir de corpus pré- étiquetés.

Dans ce chapitre, nous allons exposer la méthode de classification des textes, à travers la représentation des textes et les techniques de classification.

2.2. Représentation des textes

La représentation des textes (ou documents) est l'une des techniques qui sont utilisées pour réduire la complexité des documents et pour les rendre plus faciles a manipulé, le document est alors transformé de sa version textuelle en une matrice [Document × Terme] comme présenté dans la figure 2.1. La représentation du document la plus utilisée est le modèle appelé vectoriel dans lequel les documents sont représentés par des vecteurs de termes.

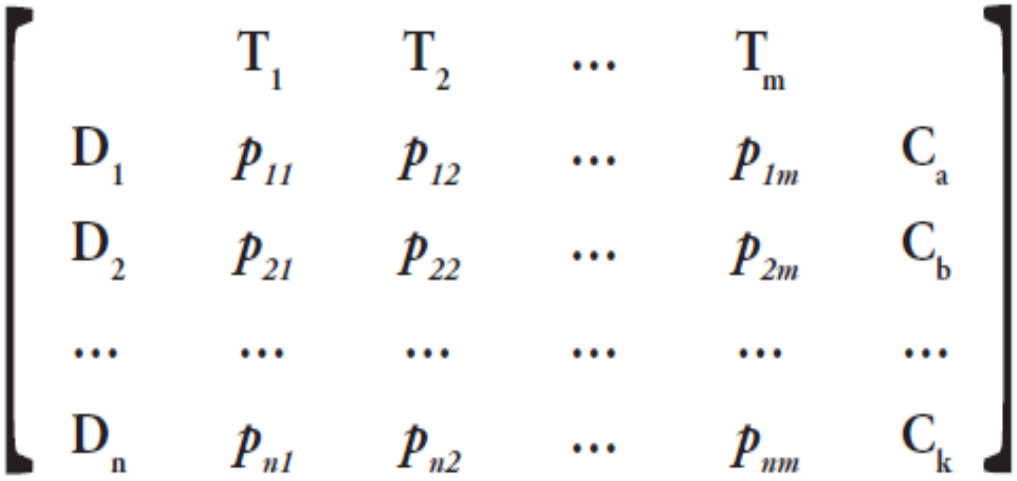


Figure 2.1 Matrice Document × Terme [14]

Chaque entrée représente un vecteur de termes ou Pnm est le poids du terme Tm dans le document Dn et Ci est la classe attribuée au document Di. [14]

2.2.1. Représentation des données textuelles

Pour représenter les documents textuels, plusieurs méthodes sont utilisées, ci-après nous présentons quelques unes :

2.2.1.1. Sac de mots

Dans cette représentation, les termes sont les mots qui constituent un texte. Dans les langues comme le français ou l'anglais, les mots sont séparés par des espaces ou des signes de ponctuations; ces derniers, tout comme les chiffres, sont supprimés de la représentation.

Les composantes des vecteurs peuvent être une fonction de l'occurrence des mots dans le texte. Cette représentation exclue toute analyse grammaticale et toute notion de distance entre les mots, et c'est pourquoi elle est appelée « sac de mots ». [15] Avec cette approche, les documents sont représentés par des vecteurs de dimension égale à la taille du vocabulaire, qui est en générale assez grande. En effet, même des collections de documents de taille moyenne peuvent contenir de nombreux mots différents, et des vocabulaires de plusieurs dizaines de milliers de mots sont désormais communs. Or la grande dimension de ces données rend la plupart des algorithmes de classification difficiles à utiliser. A cette difficulté algorithmique vient s'ajouter le fait que les représentations des données textuelles sont typiquement creuses.

2.2.1.2. Groupe de mots ou phrases

Certains auteurs proposent d'utiliser les groupes de mots comme unité de représentation. Les groupes de mots sont plus informatifs que les mots simples, car ils ont l'avantage de conserver l'information relative à la position du mot dans le groupe de mots. Par exemple «recherche d'information», «world wide web», ont un degré plus petit d'ambiguïté que les mots constitutifs. [16]

2.2.1.3. Racine ou lemme

Dans le modèle de la représentation en sac de mots, chaque flexion du mot est considérée comme un terme différent et donc une dimension de plus, ainsi que les différentes formes d'un verbe qui constitue autant de mots. Par exemple : *enseigner, enseignement, enseignant, enseignée, enseignés, enseignera,* etc. Ces mots sont considérés comme des termes différents, alors qu'il s'agit de la même racine *enseigne*. Pour la recherche des racines lexicales, il existe plusieurs algorithmes, un des plus connus pour la langue anglaise est l'algorithme de Porter.

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme singulière.

La lemmatisation est donc plus compliquée à mettre en œuvre que la recherche de racines, puisqu'elle nécessite une analyse grammaticale des textes. Un algorithme efficace, nommé TreeTagger a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise les arbres de décision pour effectuer l'analyse grammaticale, avec des fichiers de paramètres spécifiques à chaque langue. [17]

2.2.1.4. N-grammes de caractères

Cette méthode de représentation de documents texte consiste à partager ce dernier en séquences de n caractères. En effet, si nous considérons seulement les lettres de l'alphabet comme caractères et dans le cas ou « n » égal à 1, c'est-à-dire la séquence contient juste une seule lettre, est ce que c'est une bonne manière de représenter un document dans le but de le classifier ? Certainement non, même si dans certains cas cela semble très efficace, notamment dans la reconnaissance de la langue. C'est pour cette raison d'ailleurs que le « n » est toujours supérieur à 1.

Par exemple, pour générer tous les 5-grammes dans la phrase "Je suis un génie", on obtient : je_su, e_sui, suis_, _suis, uis_u, etc.

Dans la littérature, le consensus s'est porté sur n=3, car n<3 la représentation est très élémentaire, tandis que n>3 on génère beaucoup de colonnes. Les trigrammes de caractères semblent très efficaces et sont utilisés dans plusieurs applications.

2.2.2. Pondération des termes (fréquences)

Une fois que l'on choisit les composantes du vecteur représentant les *documents* $d_i \in D$, il faut décider de la façon d'associer un poids à chaque coordonnée de leurs vecteurs $\overrightarrow{di}$.

De nombreuses solutions ont été proposées dans la littérature pour coder les composantes des vecteurs, c'est-à-dire pour attribuer un poids P_{ij} à chaque terme. Ces méthodes sont basées sur les informations suivantes :

- Plus le terme t_j est fréquent dans un document d_i , plus il est en rapport avec le sujet de ce texte.

- Plus le terme t_j est fréquent dans la collection, moins il sera utilisé comme discriminant entre textes.

2.2.2.1. Pondération booléenne

Cette pondération est la plus simple représentation des données textuelles. Dans cette pondération le poids P_{ij} vaut 1 si le *terme* t_j apparaît au moins une fois dans le *document* d_i, sinon il vaut 0.

En considérant les trois phrases [18] comme trois documents :

– (d_1) : *Le chat mange la souris qui n'a pas eu le temps de manger son fromage.*

– *(d_2) : Le chat n'a plus faim et va rejoindre les autres chats.*

– *(d_3) : Le chien aboie après les chats et les souris mangent le fromage.*

Pour chaque document, une représentation vectorielle basée sur les lemmes sera utilisée avec élimination de mots vides.

Le Tableau 2.1 montre une représentation matricielle croisant en ligne les documents et en colonnes, les lemmes.

	aboyer	**chat**	**chien**	**faim**	**Fromage**	**manger**	**rejoindre**	**souris**	**Temps**
d1	0	1	0	0	1	1	0	1	1
d2	0	1	0	1	0	0	1	0	0
d3	1	1	1	0	1	1	0	1	0

Table 2.1 Exemple d'une représentation vectorielle booléenne [18]

La matrice indique uniquement la présence (1) ou l'absence (0) du lemme dans le document.

2.2.2.2. Pondération fréquentielle

Elle prend en compte le nombre d'occurrences d'un terme dans un texte. Cette mesure repose sur l'idée que plus un terme apparaît dans un texte, plus il est important. En reprenant l'exemple précédent, nous obtenons la représentation du Tableau 2.2 ci-dessous :

	aboyer	chat	chien	faim	Fromage	manger	rejoindre	souris	Temps
d1	0	1	0	0	1	2	0	1	1
d2	0	2	0	1	0	0	1	0	0
d3	1	1	1	0	1	1	0	1	0

Table 2.2 Exemple d'une représentation vectorielle fréquentielle [18]

Une telle présentation est généralement normalisée afin d'éviter de défavoriser les documents les plus longs, contenant ainsi plus de termes. La fréquence du terme t_j dans le document d_i peut être calculée par la formule suivante:

$$TF(tj, di) = \frac{\#(tj,di)}{\sum_{k=1}^{p} \#(tk,di)} \qquad (1)$$

Où: $\#(t_j , d_i)$ correspond au nombre d'occurrences du terme t_j dans d_i .

2.2.2.3. Pondération TFIDF[5]:

Elle a été introduite dans le cadre du modèle vectoriel, elle donne beaucoup d'importance aux mots qui appariassent souvent à l'intérieur du même texte, ce qui correspond bien à l'idée intuitive que ces mots sont plus représentatifs. Mais sa particularité est qu'elle donne également moins de poids aux mots qui appartiennent à plusieurs textes; pour refléter le fait que ces mots ont un faible pouvoir de discrimination entre les classes.

Cette pondération issue du domaine de la recherche d'informations tire son inspiration de la loi de Zipf introduisant le fait que les termes les plus informatifs d'un corpus ne sont pas ceux apparaissant le plus dans ce corpus. Ces mots sont la plupart du temps des mots outils. Par ailleurs, les mots les moins fréquents du corpus ne sont également pas les plus porteurs d'informations. Le poids d'un terme t_j dans un document d_i est calculé comme suit :

$$TFIDF(tj, di) = TF(tj, di) \text{ x } log \frac{N}{DF(tj)} \qquad (2)$$

[5] (Term Frequency-Inverse Document Frequency)

où $TF(t_j, d_i)$ correspond à la fréquence du terme t_j dans le document d_i ; N le nombre total des documents d'apprentissage et $DF(t_j)$ le nombre de documents contenant le terme t_j . Une telle représentation avec l'exemple précédent est donnée dans le Tableau 2.3

	aboyer	**chat**	**chien**	**faim**	**fromage**	**manger**	**rejoindre**	**souris**	**Temps**
d1	0	0	0	0	0.18	0.35	0	0.18	0.48
d2	0	0	0	0.48	0	0	0.48	0	0
d3	0.48	0	0.48	0	0.18	0.18	0	0.18	0

Table 2.3 Exemple d'une représentation TFIDF [18]

La fonction TFIDF a démontré une bonne efficacité dans des tâches de catégorisation de textes, et, en plus, son calcul est simple.

Le codage TFIDF ne corrige pas la longueur des documents. Pour ce faire, le TFIDF est normalisé. On corrige les longueurs des textes par la normalisation en cosinus, pour ne pas favoriser les documents les plus longs.

$$TFC(tj, di) = \frac{TF(tj,di) - IDF(tj,di)}{\sqrt{\sum_{k=1}^{p}(TF-IDF(tk,di))^2}} \quad (3)$$

2.3. Techniques de classification

2.3.1. Techniques d'apprentissage automatique

La notion d'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation des méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage et de remplir des tâches dont il est difficile ou impossible de remplir par des moyens algorithmiques classiques.

L'apprentissage automatique englobe toute méthode permettant de construire un modèle de la réalité à partir de données, soit en améliorant un modèle partiel ou moins général, soit en créant complètement le modèle. [19]

Nous distinguons différents types d'apprentissage : apprentissage non supervisé et apprentissage supervisé :

2.3.1.1. Apprentissage non supervisé

Dans ce type d'apprentissage, il n'existe pas de classes prédéfinies, le but est d'effectuer les meilleurs regroupements possibles, entre les objets dans lesquels, les observations diffèrent très peu, au regard de ses valeurs.

Le plus connu des problèmes non-supervisés est la classification non-supervisée ou clustering. Les classes qu'on appellera clusters, sont formées par regroupement des données qui ont certaines caractéristiques en commun.

Pour construire un regroupement de ces données, nous avons trois choix à faire :

- Choisir une mesure de ressemblance (ou similarité) entre les données

- Choisir le type de structures que nous voulons obtenir : partition, hiérarchie,arbre…

- Choisir la méthode permettant d'obtenir la structure désirée.

2.3.1.2. Apprentissage supervisé :

Il relève d'une démarche inductive consistant à construire automatiquement un classifieur qui apprend, à partir des exemples déjà classés (ou étiquetés), les caractéristiques et les propriétés des catégories cibles. Ce type d'apprentissage est dit supervisé par ce que la fonction de classification s'entraine sur les catégories (ou classe) ainsi que sur leurs caractéristiques.

La classification supervisée cherche à prédire l'appartenance de documents à des classes connues à priori. Ainsi, c'est l'ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe. [20]

Dans ce mémoire nous nous intéressons qu'à l'apprentissage supervisé dans le cas de la catégorisation de textes qui consiste à apprendre à partir d'un ensemble d'exemples une fonction de prédiction. Cette fonction permettra par la suite de prédire la classe (ou la catégorie) de chaque nouveau cas (ici le texte). La Figure 2.2 présente le principe de l'apprentissage supervisé dans le cas de filtrage de spam.

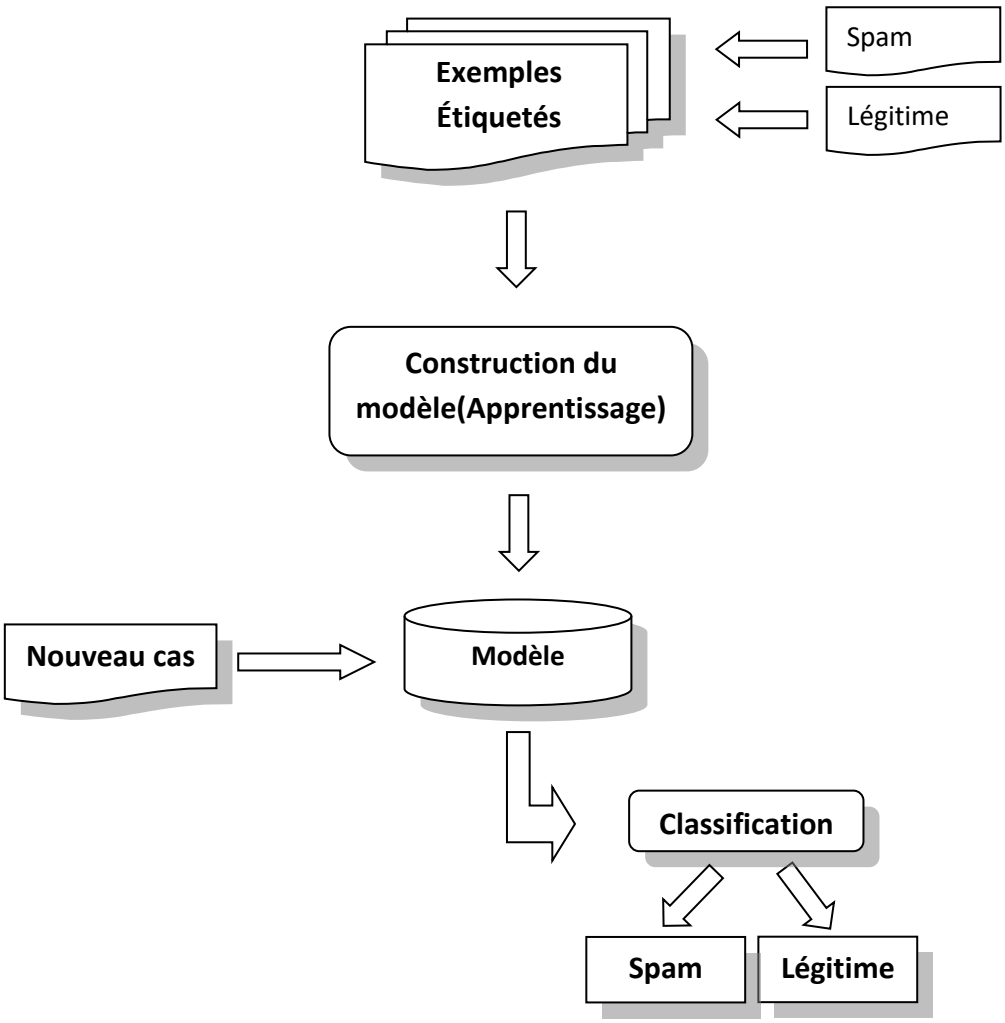


Figure 2.2 Filtrage de spam à base d'apprentissage supervisé

2.3.2. Algorithmes d'apprentissage supervisé

Dans le courant de l'apprentissage supervisé, différents types de classifieurs ont été mis au point, dans le but d'atteindre un degré maximal de précision et d'efficacité, chacun ayant ses avantages et ses inconvénients. Parmi les algorithmes d'apprentissage supervisé existants, on peut citer :

2.3.2.1. L'algorithme de Rocchio

L'algorithme de Rocchio [21] est un des plus anciens algorithmes de classification et l'un des plus simples. Un profil prototypique [c] est calculé pour chaque classe $c \in \{c_1 , c_2 \ldots c_m \}$ selon

$$[c] = \frac{\alpha}{N_c} \sum_{di \in C} di - \frac{1-\alpha}{N_{\bar{c}}} \sum_{dj \notin c} dj \qquad (4)$$

Où N_c est le nombre de documents dans c et $N\bar{c}$ le nombre de documents n'appartenant pas à c, et α un paramètre du modèle compris entre 0 et 1.

Ces profils correspondent au barycentre des exemples (avec un coefficient positif pour les exemples de la classe et négatif pour les autres). Ces vecteurs sont également normalisés de la même façon que les documents. Le classement de nouveaux documents s'opère en calculant la distance euclidienne entre la représentation vectorielle du document et celle de chacune des classes ; le document est assigné à la classe la plus proche.

L'algorithme de Rocchio est considéré comme un algorithme ancien, mais il existe des travaux qui montré que cet algorithme obtient d'excellents résultats pour la catégorisation de textes, à condition d'utiliser un codage efficace, de bien choisir les documents non pertinents et d'effectuer une optimisation des poids. Leur conclusion va à l'encontre d'autres comparaisons qui montrent que cet algorithme n'est pas performant par rapport aux méthodes fondées sur l'apprentissage numérique.

2.3.2.2. L'algorithme Naïve Bayes

Le modèle probabiliste Naïf de Bayes (NB) qui est le représentant le plus populaire des classifieurs probabilistes, est fondé sur le théorème de Bayes. [22]

Ce modèle vise à estimer la probabilité conditionnelle d'une catégorie sachant un document et affecte au document la (ou les) catégorie(s) la (les) plus probable(s). La partie naïve de ce modèle est l'hypothèse d'indépendance des mots, c'est-à-dire que la probabilité conditionnelle d'un mot sachant une catégorie est supposée indépendante de cette probabilité pour les autres mots.

Considérons $\overrightarrow{di}$ = (w_{i1},w_{i2} ,……. ,w_{ip}) la représentation vectorielle représentant un texte d_i et C = {c_1,c_2 ,….,c_m} un ensemble de classes. En s'appuyant sur le théorème de

Bayes, la probabilité que ce document appartienne à la classe c_k dans notre cas C_k ={spam,ham} est définie par :

$$P(C_k/di) = \frac{P(C_k) \text{x } P(di/C_k)}{P(di)} \qquad (5)$$

Le but étant de discriminer les différentes classes, il suffit donc d'ordonner $P(C_k/d_i)$ pour toutes les classes. On peut alors supprimer le dénominateur $P(d_i)$ qui est le même pour toutes les classes. $P(C_k)$ est la probabilité à priori qui est estimée par le pourcentage d'exemples appartenant à la classe C_k dans le corpus d'apprentissage.

En faisant l'hypothèse que les termes sont indépendants, la probabilité conditionnelle $P(d_i / c_k)$ est définie par :

$$P(di/C_k) = \prod_{j=1,p} P(Wij/C_k) \qquad (6)$$

La classe c_k d'appartenance de la représentation vectorielle $\overrightarrow{di}$ d'un document *di* est définie par :

$$C_{NB} = \arg\max P(C_k) \prod_j P(Wij/C_k) \qquad (7)$$

En d'autres termes, le classifieur Naïve Bayes affecte au document d_i la classe ayant obtenu la probabilité d'appartenance la plus élevée.

2.3.2.3. Les K voisins les plus proches

Cette méthode a prouvé son efficacité face au traitement des données textuelles. La phase d'apprentissage consiste à stocker les exemples étiquetés. Le classement de nouveaux textes s'opère en calculant la similarité[6] entre la représentation vectorielle du document et celle de chaque exemple du corpus d'apprentissage. Les k éléments les plus proches sont sélectionnés

[6] la mesure de similarité la plus couramment utilisée est le calcul du cosinus de l'angle formé par les deux vecteurs de documents.

et le document est assigné à la classe majoritaire (le poids de chaque exemple dans le vote étant éventuellement pondéré par sa distance).

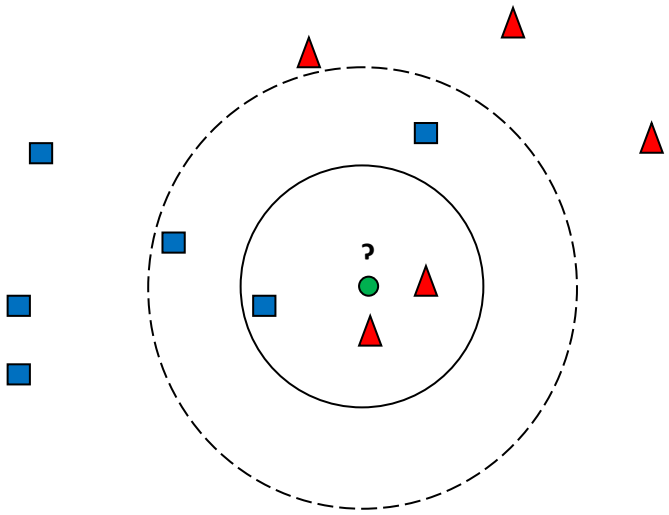


Figure 2.3 K-ppv dans un espace à deux dimensions

Le choix de la valeur de k est dépendant de la taille de l'échantillon et des classes, et influence les résultats de la classification. Dans l'exemple de la Figure 2.3, l'objet rond sera classifié triangle si k=3 et classifié Carré si k=5.

Lorsque « k » est petit, la classification est plus sensible à cause des documents appartenant à une classe mais dont leur vecteur de représentation ressemble beaucoup plus à une autre. Par contre, lorsque « k » est trop grand, les catégories ayant peu d'exemples peuvent être désavantagées par rapport à celles qui en ont plus. On peut remédier à cela en pondérant le vote par la distance qui sépare les plus proches voisins de l'individu à classer.

Si la qualité de catégorisation obtenue par les k plus proches voisins (K-ppv) est satisfaisante que celle obtenue avec d'autres méthodes qui nécessitent un apprentissage complexe, le temps nécessaire à son déroulement peut être un obstacle difficilement incontournable ; là où la complexité des autres méthodes est fonction du nombre de catégories.

2.3.2.4. Les machines à support vectoriels

Le but de SVM est de trouver un classificateur qui sépare au mieux les données et maximise la distance entre ces deux classes. Ce dernier est un classificateur linéaire appelé hyperplan. Comme montré dans la Figure 2.4, cet hyperplan sépare les deux ensembles de points.

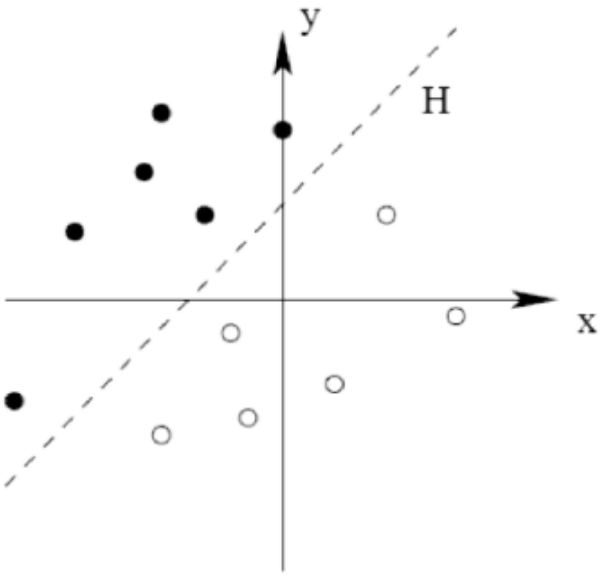


Figure 2.4 La séparation du l'hyper plan par les SVM [23]

Les points les plus proches, qui seuls sont utilisés pour la détermination de hyperplan, sont appelés vecteurs de support (voir Figure 2.5).

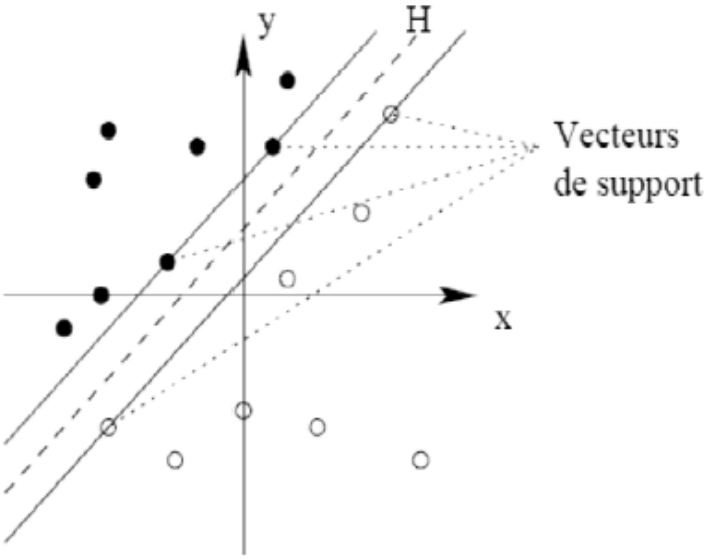


Figure 2.5 Les vecteurs de support [23]

2.3.2.5. Les règles de décision

Ce classifieur consiste à construire pour chaque catégorie c_i un ensemble de règles, permettant de distinguer une catégorie d'une autre.

Une règle est de type SI <prémisse> alors <catégorie> où la prémisse est en forme normale disjonctive (F.N.D). Les littéraux dans les prémisses représentent la présence ou l'absence d'un terme dans le document de test d_k , tandis que la clause de la règle, indique la décision pour classer d_k dans la catégorie c_i.

Les systèmes d'apprentissage de règles varient considérablement en termes de méthodes, d'heuristiques et de critères employés pour la généralisation et l'élagage de règles.

2.4.3. Remarques sur les algorithmes d'apprentissage supervisé

• *Le classifieur naïf de Bayes* s'est montré très performant pour des tâches de classification de textes comme le filtrage de spam que d'autres méthodes malgré son hypothèse d'indépendance des mots. Il reste capable de bien fonctionner avec des données incomplètes comme il peut être appliqué à de nombreux secteurs d'activité: médicale, juridique, etc.

• *Les SVM* donnent de très bons résultats de classification de textes mais sont très coûteux en temps d'apprentissage et possèdent une limitation théorique, Le modèle sous-jacent à ce classifieur a été conçu pour la classification binaire : il cherche un plan séparateur qui sépare l'ensemble des objets en deux classes.

• *Les K-ppv* sont très simples à mettre en œuvre, et permettent une implémentation rapide pour fournir des résultats satisfaisants. Cette méthode reste robuste sur des cas de données incomplètes, mais elle est très couteuse en temps de classification et stockage mémoire.

2.4. Conclusion

Dans ce chapitre, nous avons exposé la tache de classification des textes, à travers la représentation des textes et les techniques de classification, finalement nous avons présenté quelques remarques sur les algorithmes d'apprentissage supervisé.

CHAPITRE 3

TRAVAUX PUBLIES SUR LE FILTRAGE DE SPAM

3.1. Introduction

Ce chapitre présente quelques techniques de filtrage de spam basées sur l'apprentissage supervisé.

3.2. Drucker et al.

Drucker et al. [24] ont comparé l'efficacité du classifieur linéaire SVM avec ceux de RIPPER, Rocchio et arbres de décision. Il est la première qui a essayé un large ensemble de configurations d'expérimentations sur la sélection des termes et les différents algorithmes d'apprentissage. Ils arrivent aux conclusions suivantes :

- SVM (avec une représentation binaire) et arbres de décision (avec une représentation TF) sont les deux meilleurs classifieurs, mais les SVM permettent d'atteindre des taux de faux positifs plus bas et plus facilement.
- Dans un choix entre l'utilisation d'une liste de stopwords ou non, il est préférable qu'une liste de stopwords ne soit pas utilisée.
- L'apprentissage en utilisant les arbres de décision est énormément long.
- Les méthodes RIPPER et Rocchio ne sont pas performantes pour le filtrage de spam.

3.3. Saumya Goyal et al.

En 2016, Saumya Goyal et al. [25] proposent l'utilisation d'un mécanisme de détection de spam basé sur l'algorithme KNN et l'arbre de décision, ils appliquent ces algorithmes sur des ensembles de données réels de twitter. Pour analyser le mécanisme proposé l'outil WEKA[7] est utilisé. Les mesures de performance telles que TP Rate, FP Rate, Precision, Recall et F-Measure sont utilisées pour évaluer le mécanisme proposé.

Ils obtenaient les résultats suivants :

[7] Waikato Environment for Knowledge Analysis

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
KNN	1	0.508	0.904	1	0.949	Spam
	0.492	0	1	0.492	0.659	Normal
	0.912	0.42	0.92	0.912	0.899	Weighted Avg
Arbre De Décision	1	1	0.827	1	0.905	Spam
	0	0	0	0	0	Normal
	0.827	0.827	0.683	0.827	0.748	Weighted Avg

Table 3.1 Les mesures de performance avec l'arbre de décision [25]

Les résultats obtenus présente que l'algorithme KNN est plus performant par rapport a l'arbre de décision.

3.4. Nurul Fitriah Rusland et al

En 2017, Nurul Fitriah Rusland et al [26] ont testé un algorithme naïve bayes pour le filtrage du spam sur deux corpus (Spam Data qui contient 9324 e-mails et 500 attributs et SPAMBASE qui contient 4601 email et 58 attributs) et tester ses performances. L'architecture du système est comme suite :

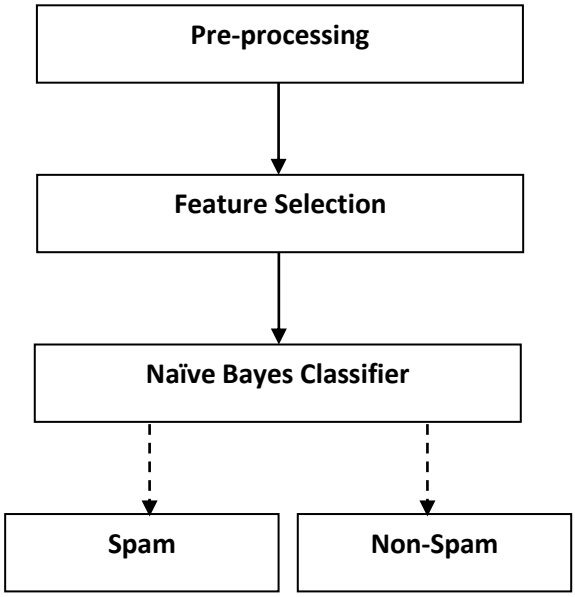
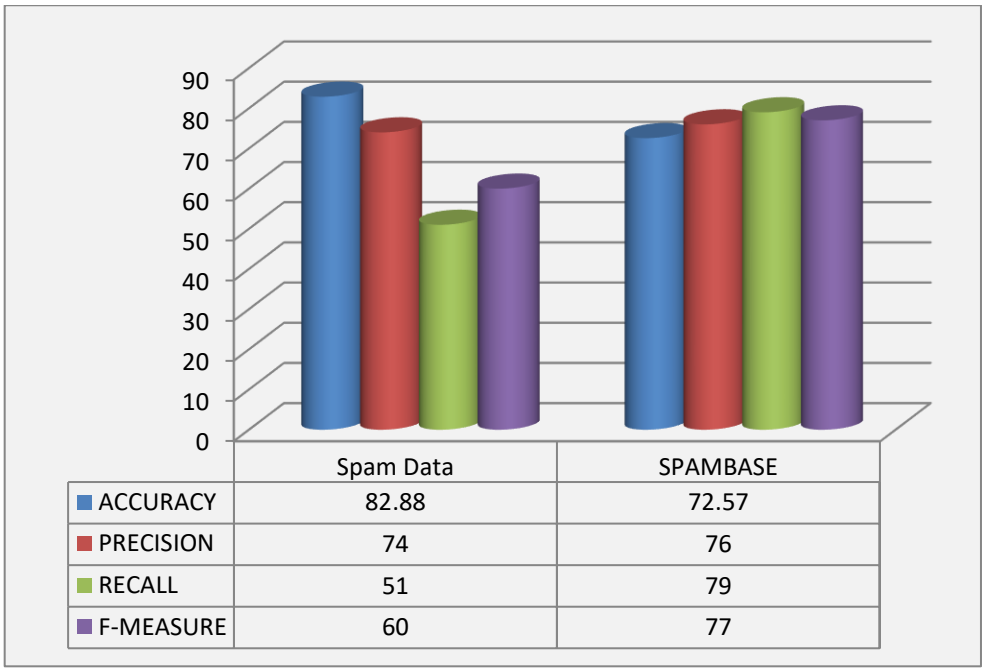


Figure 3.1 Filtre anti-spam en utilisant l'algorithme naïve bayes [26]

La performance de ce filtre est évaluée avec l'outil WEKA en fonction de leur précision, de leur rappel et de leur F-mesure. Ils obtenaient les résultats suivants :



	Spam Data	SPAMBASE
ACCURACY	82.88	72.57
PRECISION	74	76
RECALL	51	79
F-MEASURE	60	77

Figure 3.2 Résultats d'évaluation avec les deux corpus

Ils ont constaté que La performance de ce filtre est également basée sur les corpus utilisés. Comme on peut le voir, les corpus qui ont moins d'instances et d'attributs (ici SPAMBASE) peuvent donner de bons résultats.

3.5. Anju Radhakrishnan et Vaidhehi V.

Anju Radhakrishnan et Vaidhehi V. [27] utilisent deux algorithmes importants à savoir, Naïve Bayes et J48 Decision Tree et testent leur efficacité dans la classification des emails. Le corpus utilisé est Enron et la valeur TF-IDF est utilisée comme fréquence.

Les classifieurs sont également testés avec différentes tailles des attributs. Les résultats des tests sont présentés comme suite :

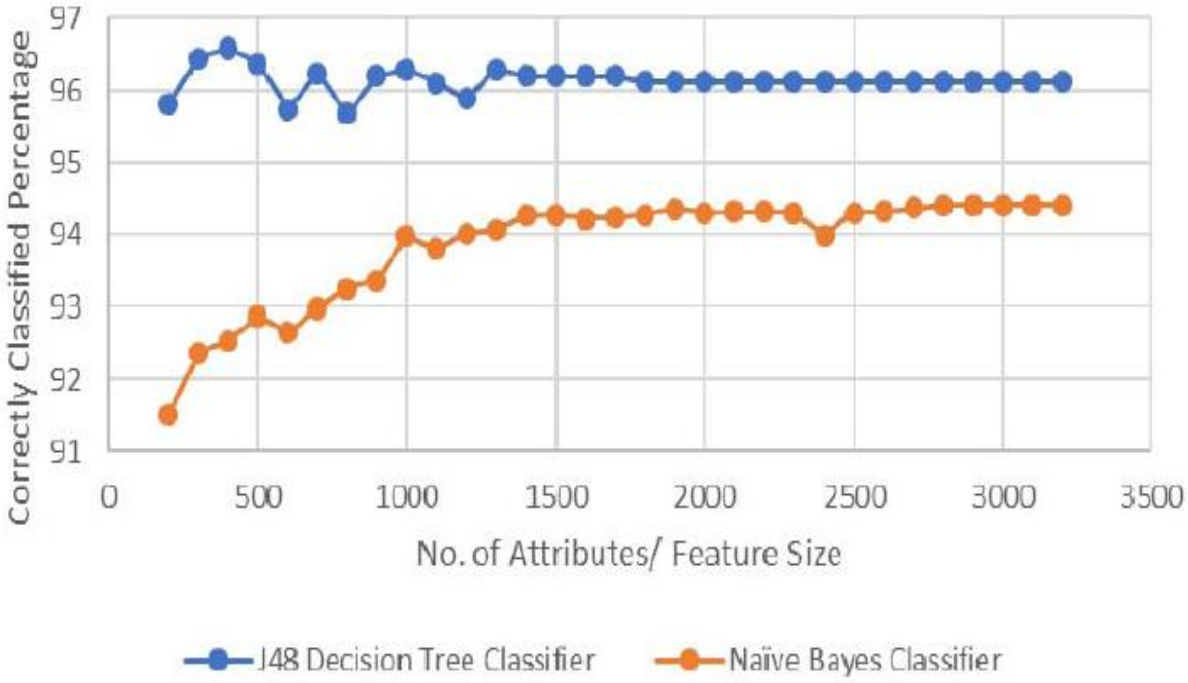


Figure 3.3 Les résultats des tests pour les deux classifieurs NB et J48 [27]

Les expériences de classification d'email a montré que le classifieur J48 Decision Tree est plus efficace que le classifieur Naïve Bayes pour le corpus Enron. Il donne une précision de 96,5971% dans la classification des e-mails avec une taille de 400 attributs.

3.6. Shradhanjali et Verma Toran

Shradhanjali et Verma Toran [28] proposent l'utilisation d'une nouvelle méthode pour la détection de spam en utilisant SVM et l'extraction des attributs qui atteint une précision de 98%. L'architecture du système proposé est présentée dans la figure suivante :

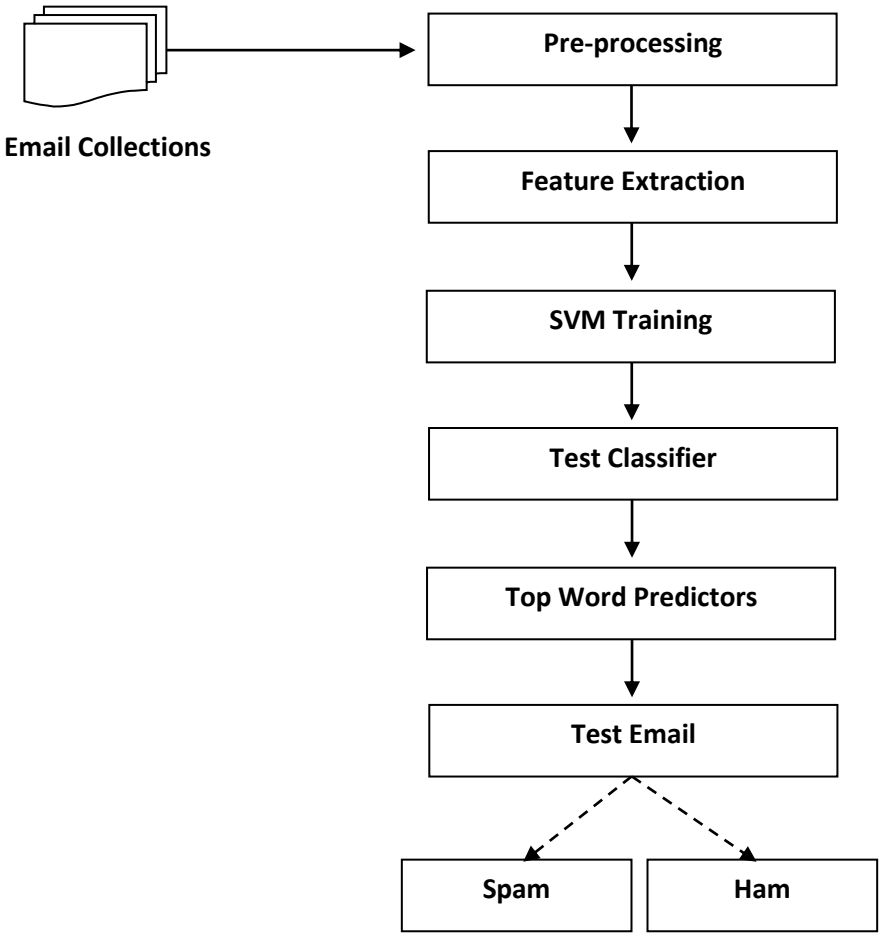


Figure 3.4 Filtre anti-spam en utilisant SVM et l'extraction des attributs [28]

Prétraitement: dans l'étape de prétraitement, tous les numéros, les symboles spéciaux, les balises URL et HTML sont supprimées. Le stemming est fait pour enlever l'alphabet inutile dans les mots.

L'extraction des attributs : Après le prétraitement, les attributs sont extraits.

Entraînement : Après l'extraction des attributs, l'entrainement est fait. Lors de l'entrainement, les e-mails sont fournis en entrée du classifieur SVM.

Test de classifieur : Après l'entrainement, les e-mails de test sont donnés pour tester l'exactitude du système. La précision est atteinte jusqu'à 98%.

Classification : Enfin, le classifieur est testé avec un e-mail (La classe spam ou la classe légitime).

3.7. Jawale Diksha .S et.al

En 2018, Jawale Diksha .S et.al [29] proposent l'utilisation d'un classifieur de spam hybride NB-SVM qui utilise les avantages de Naïve Bayes (NB) et Support Vector Machine (SVM), NB est un algorithme de classification rapide et SVM a une grande performance en raison de leur taux de rappel et de précision élevé.

Les données d'apprentissage sont d'abord traitées par l'algorithme NB dans lequel il calcule la probabilité pour chaque mot et message et compare avec un seuil qui classifie Les données. Les données traitées par NB vont à SVM pour améliorer la précision.

L'architecture de ce classifieur est comme suite :

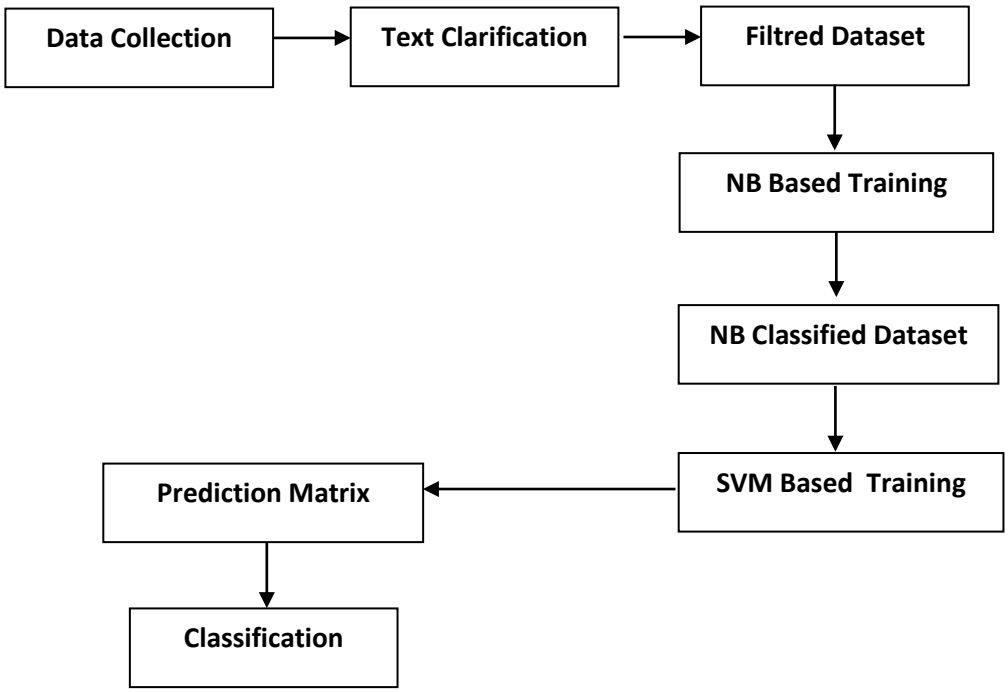


Figure 3.5 Architecture NB-SVM [29]

Avec l'utilisation de NB, ils obtiennent une précision de 96,65% dans la phase d'entraînement et 95,78% dans la phase de test. Avec SVM, ils obtiennent une précision de 99,43% dans la phase d'entraînement et 97,13% dans la phase de test. La combinaison de ces deux algorithmes NB-SVM, donne une précision de 99,44% dans la phase d'entraînement et 97,57% dans la phase de test. Ce qui montre que les résultats étaient meilleurs que ceux des deux classifieurs utilisés séparément.

3.8. Conclusion

Dans ce chapitre, nous avons présenté quelques travaux publiés sur le filtrage de spam. Nous détaillons leurs principes ou architectures et leurs résultats.

CHAPITRE 4

ÉTUDE COMPARATIVE

4.1. Introduction

En ce chapitre, nous présentons une étude comparative entre des systèmes de filtrage de spam basant sur les trois algorithmes de base (SVM, NB et KNN) en utilisant le corpus smsSpamCollection. Basant sur ces résultats, on le discute et on va sélectionner le meilleur algorithme parmi les algorithmes étudiés pour la classification des emails reçus.

4.2. Architecture du système

L'architecture de notre modèle de filtrage anti-spam est représentée sur la figure 4.1. Tout d'abord, en utilisons le corpus SMS Spam Collection qui en définit en section 4. Le corpus de messagerie va passer par la phase de préparation ou prétraitement puis en utilisant l'un des algorithmes d'apprentissage (SVM, NB, KNN) pour construit un modèle qui permet de classer les nouveaux messages.

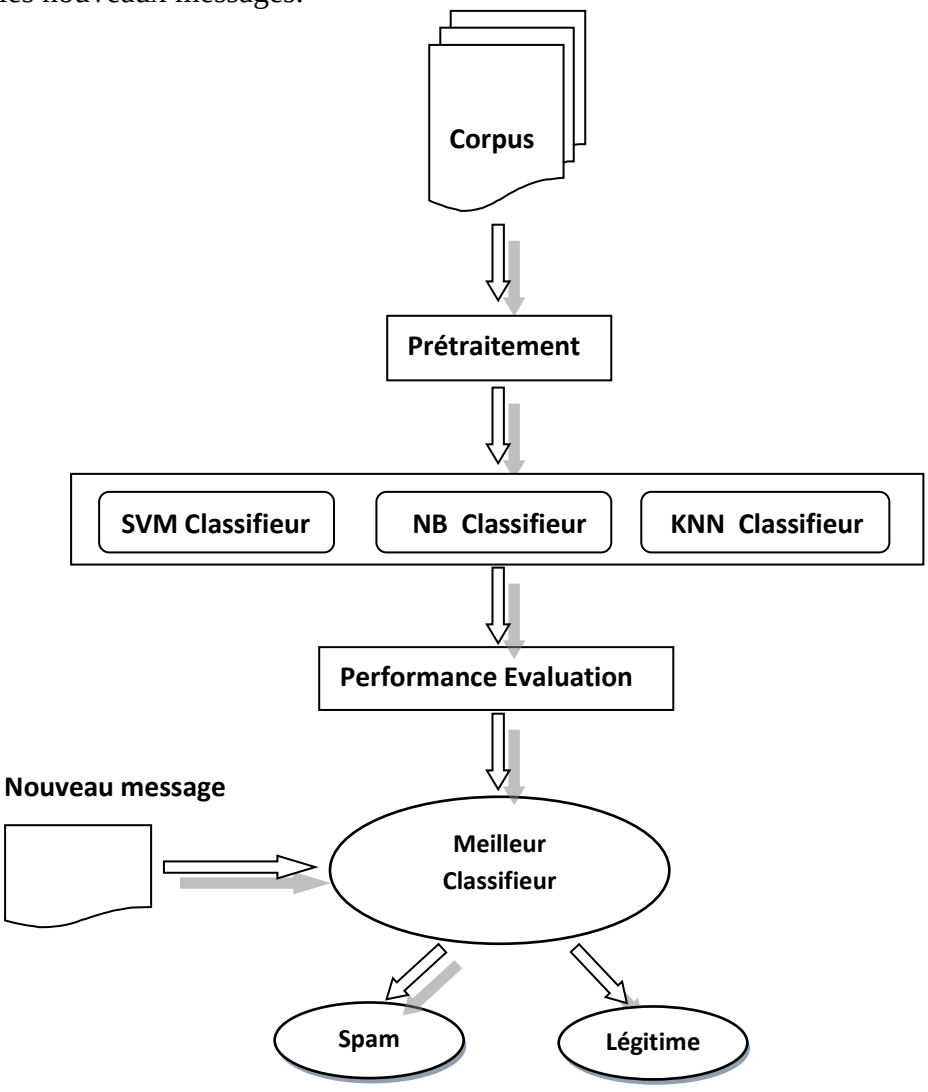


Figure 4.1 Architecture du système

4.3. Présentation des outils de développement

4.3.1. Langage JAVA

Notre choix du langage de programmation s'est porté sur le langage JAVA, et cela parce qu'il est un langage orienté objet simple ce qui réduit les risques d'incohérence et il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers ainsi que beaucoup de fonctionnalités qui peuvent être utilisé pour développer des applications diverses. Aussi, il offre un nombre important de fonctions de traitement de texte.

4.3.2. Environnement de développement

L'environnement de développement utilisé est le NetBeans 8.2, il possède de nombreux avantages qui sont les suivants :

- un environnement de développement intégré (EDI)
- Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces).
- Supports de plusieurs plateformes d'exécution : Windows, Lunix, Mac OS.

4.3.3. Weka

Nous avons effectué des expériences en intégrant quelque bibliothèques de l'outil WEKA dans notre IDE, en fait, (Waikato Enveronment Knowledge Analysis) offrent un ensemble d'algorithmes permettant de manipuler et d'analyser des fichiers de donnés. Il permet à l'utilisateur d'implémenter la plupart des algorithmes d'apprentissage entre autres : des Machine à Vecteurs de Support, Naïve Bayes, K voisins les plus proches et les arbres de décision. [30]

Il se compose principalement :

- De classe Java permettant de charger et manipuler des donnés.
- De classe Java pour implémenter les principaux algorithmes de classification supervisée et non supervisée.
- D'outils de sélection d'attributs, des statistiques sur ces attributs.
- De classes permettant de visualiser les résultats.

4.4. Description du corpus utilisé

Le corpus SMS Spam Collection v.1 [31] est un ensemble commun de messages étiquetés SMS. Il dispose d'une collection composée de 5 574 messages en anglais, réels, étiqueté selon étant légitime (Ham) ou spam. Cette collection contient 747 messages spam et 4827 messages légitimes.

Utilisation

La collection est composée d'un seul fichier texte où chaque ligne contient le message brut suivie par la bonne classe. Nous vous proposons quelques exemples ci-dessous:

```
What you doing?how are you?   ham
Ok lar... Joking wif u oni...   ham
dun say so early hor... U c already then say...   ham
MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*   ham
Siva is in hostel aha:-.   ham
Cos i was out shopping wif darren jus now n i called him 2 ask wat
present he wan lor. Then he started guessing who i was wif n he
finally guessed darren lor.   ham
FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk
time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16
stop?txtStop   spam
Sunshine Quiz! Win a super Sony DVD recorder if you canname the
capital of Australia? Text MQUIZ to 82277. B   spam
URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller
Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call
0871-872-9758 BOX95QU   spam
```

4.5. Les mesures d'évaluation

Les critères de mesure des performances sont le rappel (recall) et la précision et aussi la f-mesure qui est la combinaison des deux précédentes. [32]

Pour mesurer toutes ces métriques, nous devons tout d'abord calculer les valeurs suivantes:

- N(LL) : le nombre de courriels légitimes classifiés légitimes (Vrai négatifs).

- N(SS) : le nombre de courriels spam classifiés spam (Vrai positifs).

- N(LS) : le nombre de courriels légitimes classifiés spam (Faux Positifs).

- N(SL) le nombre de courriels spam classifiés légitimes (Faux négatifs).

Nous avons alors les mesures suivantes :

$$Precision = \frac{\mathrm{N(SS)}}{\mathrm{N(SS)+N(LS)}} \qquad (8)$$

$$Recall = \frac{\mathrm{N(SS)}}{\mathrm{N(SS)+N(SL)}} \qquad (9)$$

$$F - Measure = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision + Recall} \qquad (10)$$

4.6. Description du système

Dans cette partie nous nous intéressons à la présentation de notre système en montrant quelques exemples de captures d'écran des différentes interfaces réalisées.

Nous avons essayé de créer une interface graphique qui montre le plus possible les détails d'exécution de notre application.

4.6.1. Prétraitement

Au lancement de l'application la fenêtre suivante s'affiche :

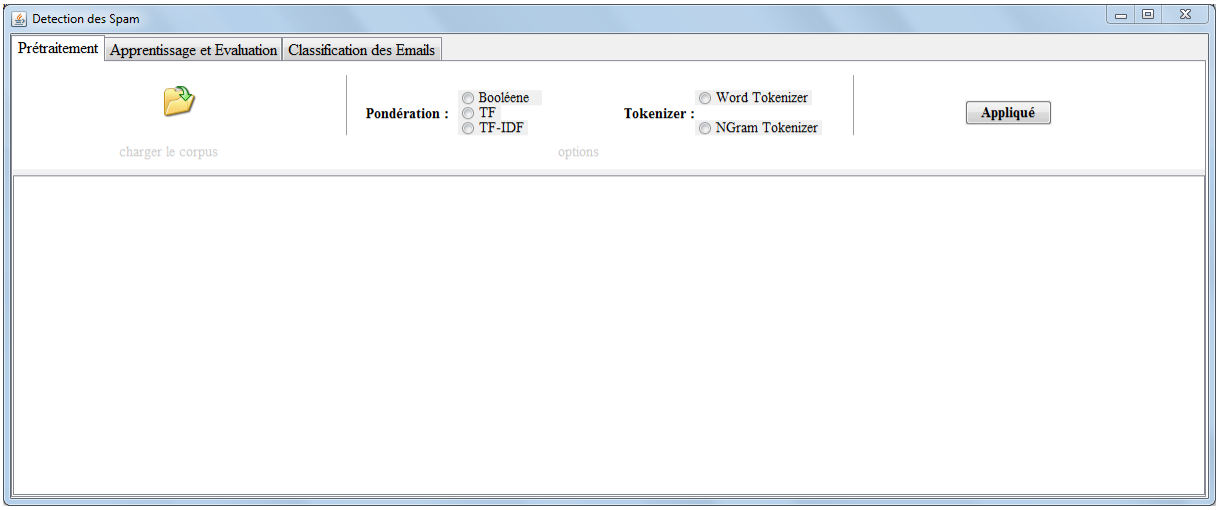


Figure 4.2 Interface prétraitement

Prétraitement est la première étape nécessaire pour travailler avec des techniques d'apprentissage automatique. Le corpus doit être sous une forme compréhensible par le système d'apprentissage. Avant d'effectuer les expériences, les e-mails doivent être prétraités. Il faut enregistrer les e-mails en tant que fichiers ARFF.

Un fichier ARFF (Attribute-Relation File Format) est un fichier texte qui décrit une liste des instances qui partagent un ensemble d'attributs. Fichiers ARFF ont deux sections distinctes. La première section est l'information d'en-tête, qui est suivi de l'information de données .L'en-tête du fichier ARFF contient le nom de la relation, la liste des attributs (colonnes dans les données), ainsi que leur type.

Le fichier ARFF dans notre cas se compose de :

- nom de la relation : @relation ' smsspam '
- la liste des attributs : @attribute Message string
  @attribute Class {spam,ham}
- la liste des instances : chaque ligne représente une description, par la liste des valeurs de chacun de ses attributs.

- chargement de corpus

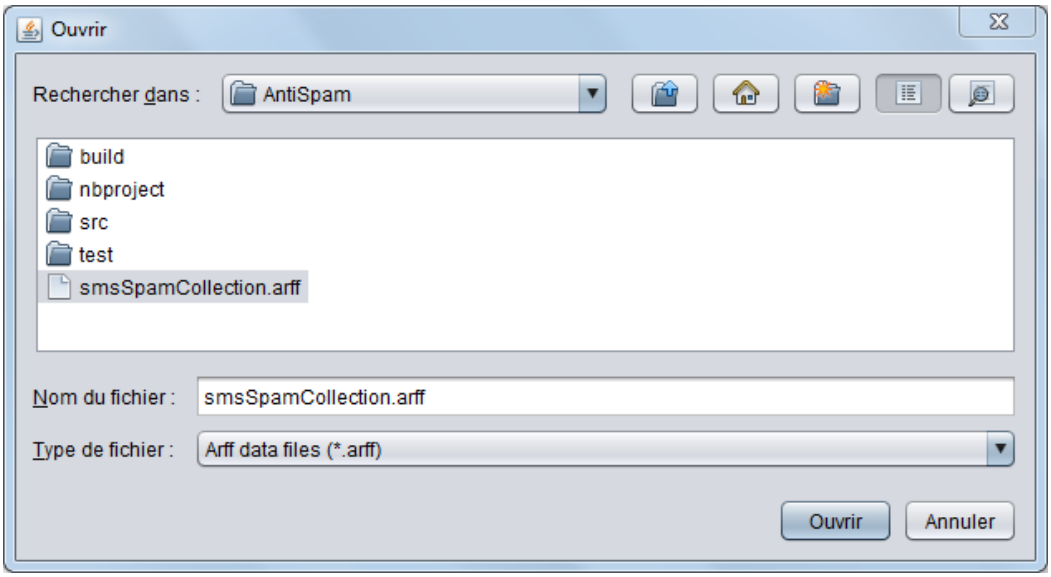


Figure 4.3 chargement de corpus

- choix des options Pondération et Tokeniser

- Pondération : en calcule la pondération de chaque token avec l'un des méthodes : booléenne, TF ou TFIDF.

- Tokenisation : décompose chaque texte de corpus en : Word Tokenizer ou NGram Tokenizer (N = {1 ; 2 ; 3}).

Figure 4.4 Choix des options

- après de fixer les options on clique sur le bouton Appliqué qui permet de construire un vecteur représente chaque texte et d'afficher les attributs. La figure 4.5 illustrer ça.

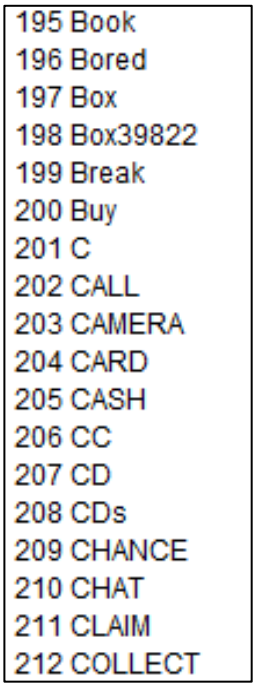


Figure 4.5 Affichage des attributs

4.6.2. Apprentissage et évaluation

Cette étape est la deuxième étape (Figure 4.6) consiste à implémenter et évaluer les trois classifieurs (SVM, NB, KNN), pour cela en a utilisé les méthodes suivant :

- buildClassifier(Instances data) ; pour construire le classifieur
- evaluateModel(Classifier classifier, Instance instance) ; pour évaluer le classifieur
- toSummaryString() ; pour afficher les résultats d'évaluation

Nous avons effectué la validation croisée c'est-à-dire pour chaque sous-ensemble, nous avons créé 10 paires de partitions d'entrainement et de validation pour examiner la performance comme une moyenne sur 10 itérations.

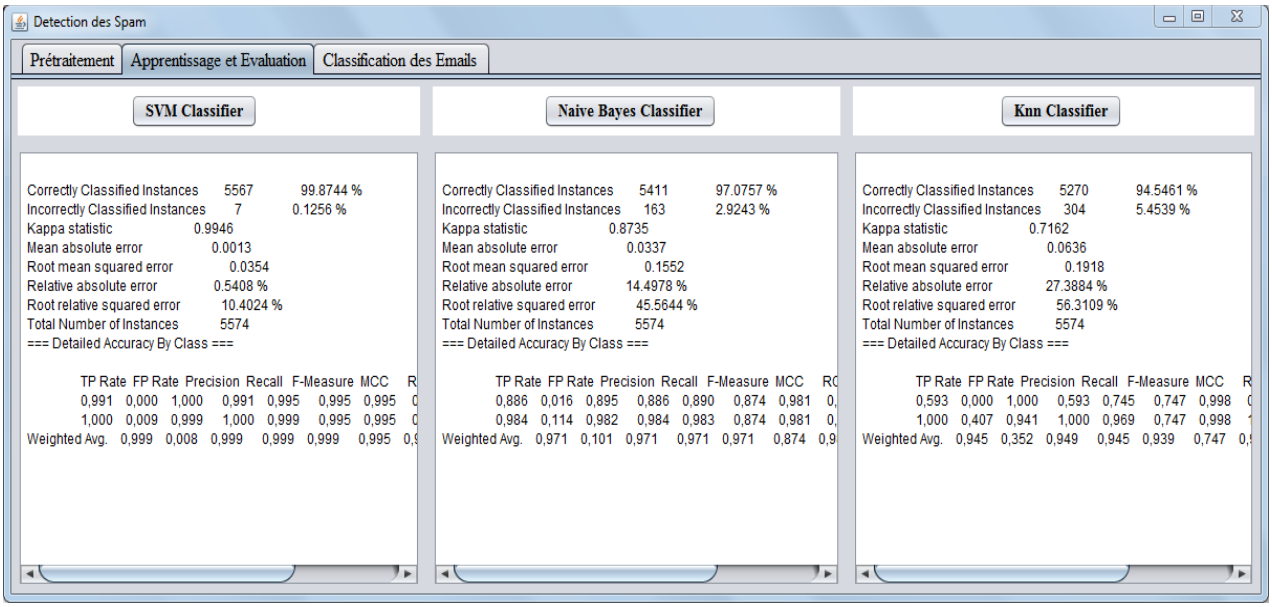


Figure 4.6 Apprentissage et évaluation

4.6.3. Classification des emails

Permet de classifier les nouveaux emails reçus en utilisant le meilleur algorithme

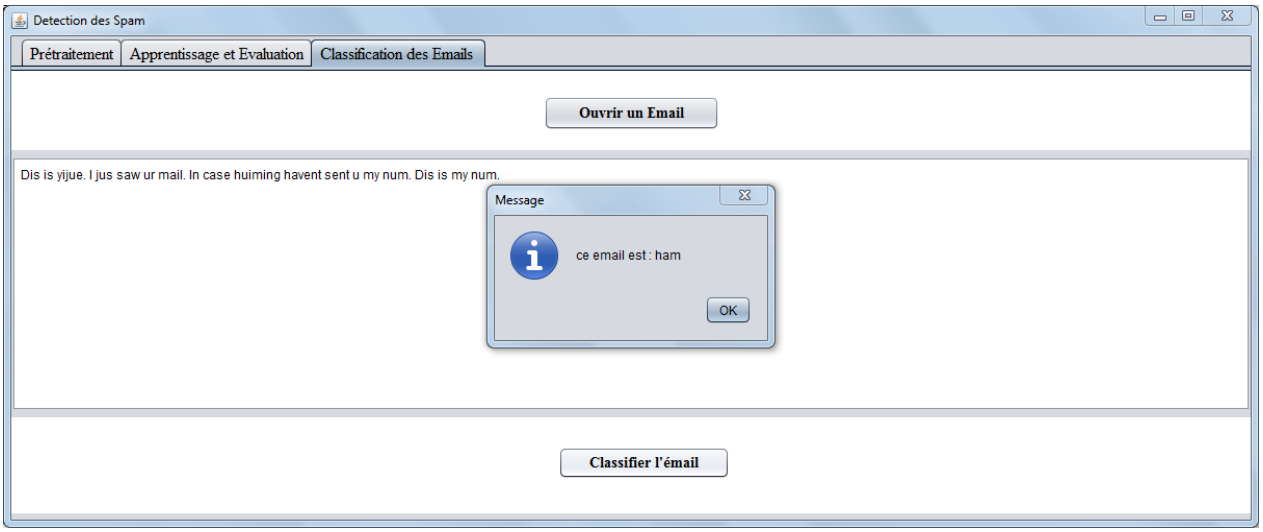
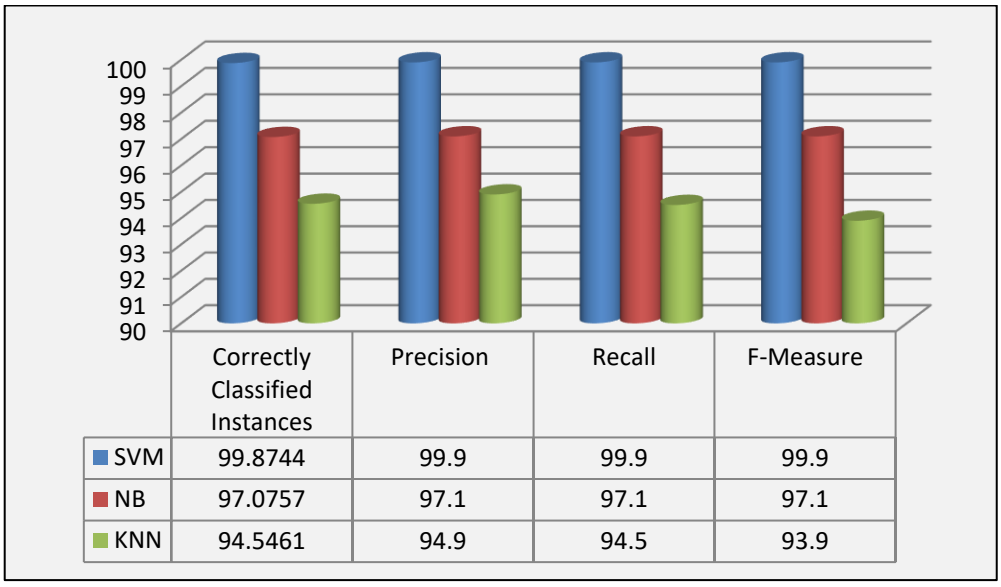


Figure 4.7 Classification des nouveaux emails

4.7. Les résultats

En à déjà vus dans la section précédente qu'en a appliquer les trois algorithmes en 6 différant cas : combinaison entre les pondérations (booléenne, TF, TF-IDF) avec (1Token, NGram Token). on a présenter les résultats avec les mesures d'évaluation Précision, Racall et F-Measure.

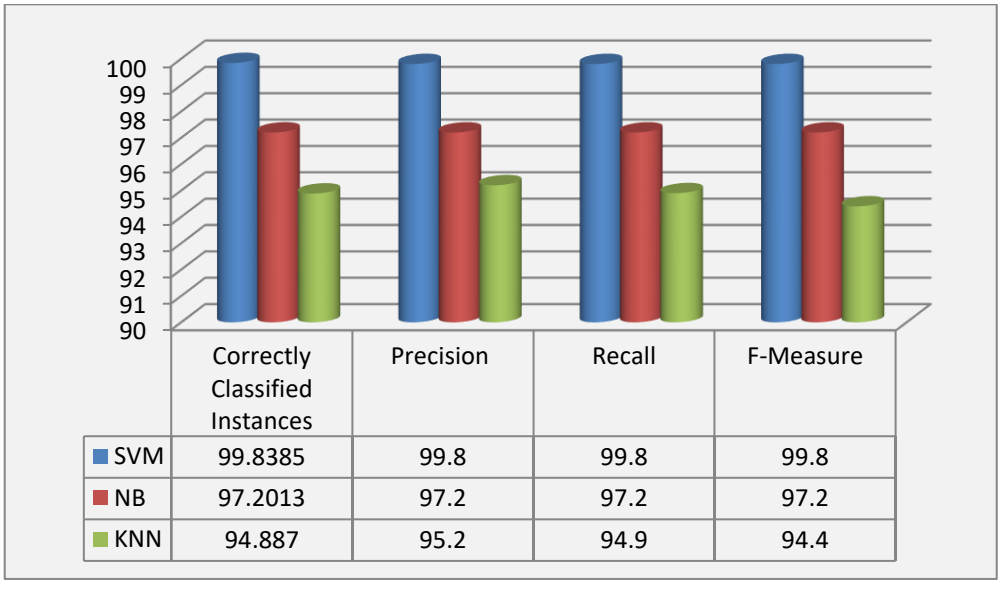
- La Figure 4.8 illustre la pondération booléenne avec 1Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.8744	99.9	99.9	99.9
NB	97.0757	97.1	97.1	97.1
KNN	94.5461	94.9	94.5	93.9

Figure 4.8 Pondération booléenne avec 1Token

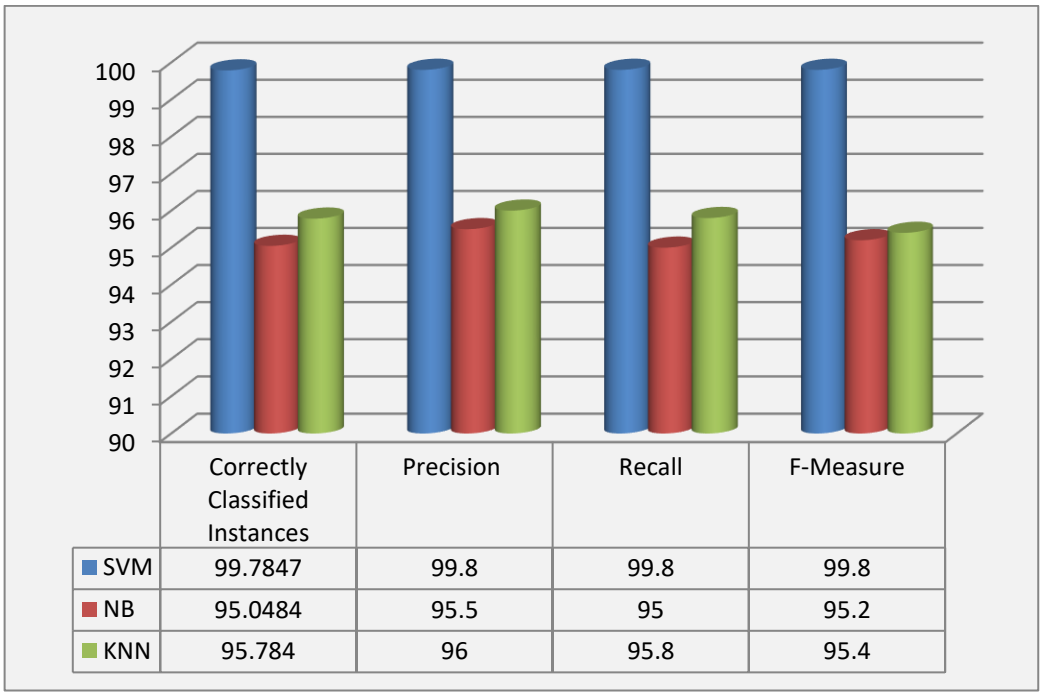
- La Figure 4.9 présente la pondération booléenne avec NGram Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.8385	99.8	99.8	99.8
NB	97.2013	97.2	97.2	97.2
KNN	94.887	95.2	94.9	94.4

Figure 4.9 Pondération booléenne avec NGram Token

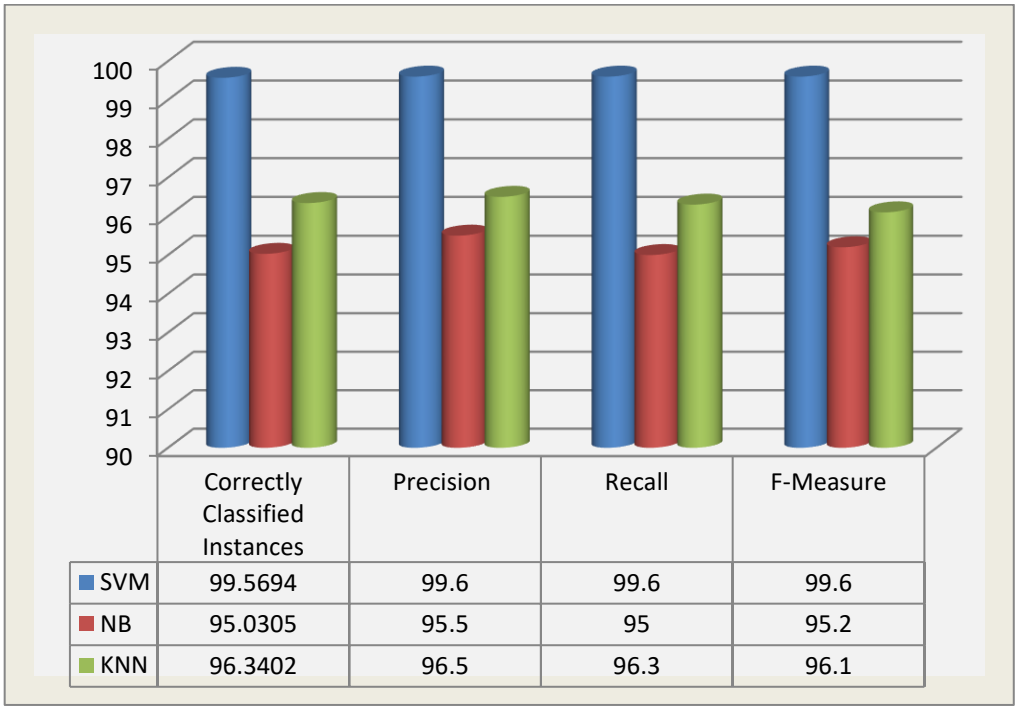
➢ La figure 4.10 présente la pondération TF avec 1Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.7847	99.8	99.8	99.8
NB	95.0484	95.5	95	95.2
KNN	95.784	96	95.8	95.4

Figure 4.10 Pondération TF avec 1Token

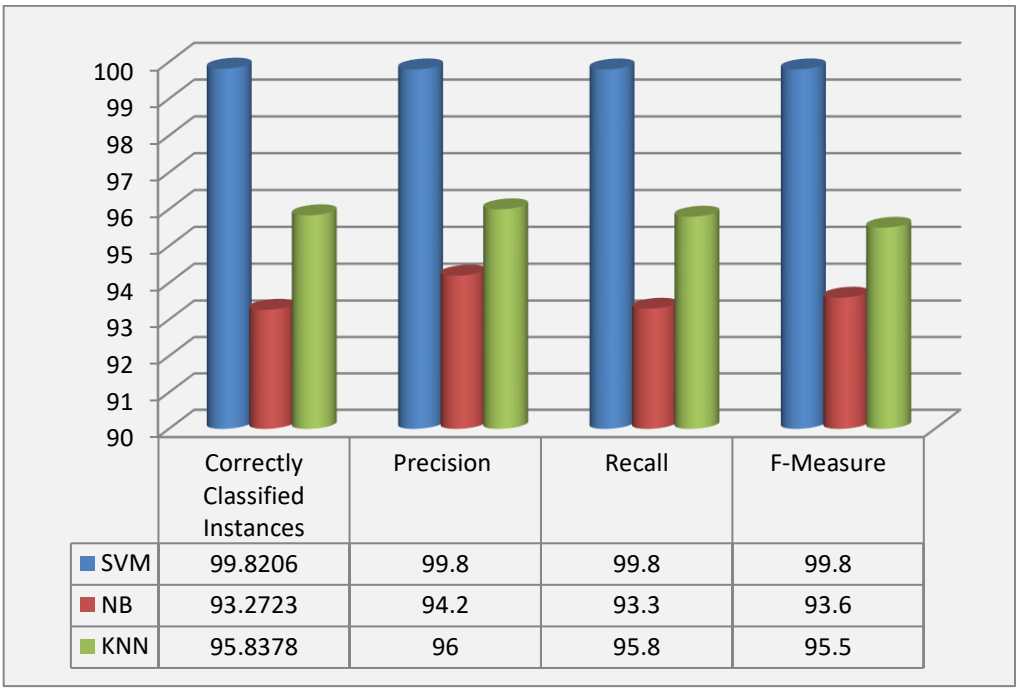
➢ La figure 4.11 illustre la pondération TF avec NGram Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.5694	99.6	99.6	99.6
NB	95.0305	95.5	95	95.2
KNN	96.3402	96.5	96.3	96.1

Figure 4.11 Pondération TF avec NGram Token

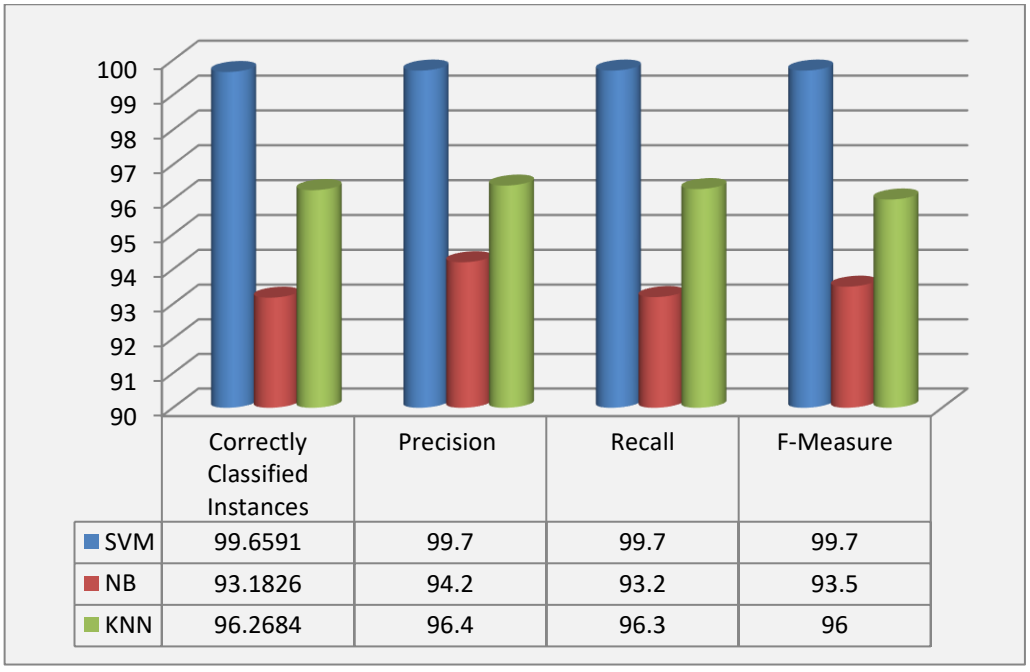
➢ La figure 4.12 présente la pondération TF-IDF avec 1Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.8206	99.8	99.8	99.8
NB	93.2723	94.2	93.3	93.6
KNN	95.8378	96	95.8	95.5

Figure 4.12 Pondération TF-IDF avec 1Token

➢ La figure 4.13 illustre la pondération TF-IDF avec NGram Token



	Correctly Classified Instances	Precision	Recall	F-Measure
SVM	99.6591	99.7	99.7	99.7
NB	93.1826	94.2	93.2	93.5
KNN	96.2684	96.4	96.3	96

Figure 4.13 Pondération TF-IDF avec Ngram Token

4.8. Discussion

Les résultats de notre application sont présentés sur les graphiques précédentes pour les trois algorithmes de classification SVM, NB et KNN.

Grace aux résultats obtenus en observe que :

- l'algorithme SVM atteindre une précision et un rappel de 99.9 comme meilleur résultat, avec la pondération booléenne et 1Token.
- l'algorithme NB atteindre une précision et un rappel de 97.2 comme meilleur résultat, avec la pondération booléenne et NGram Token.
- l'algorithme KNN atteindre une précision de 96.5 et un rappel de 96.3 comme meilleur résultat, avec la Pondération TF et NGram Token.

En observe aussi que pour la pondération booléenne l'algorithme NB donne des bons résultats par rapport à l'algorithme KNN, Contrairement aux Pondérations TF et TF-IDF.

L'algorithme SVM est le plus performant par apport aux algorithmes NB et KNN, Les graphes obtenus démontre ça. Et donc on va l'utiliser dans la classification des nouveaux emails reçus.

4.9. Conclusion

Dans ce chapitre on a présente l'architecture de notre système de filtrage des spams et ainsi une vue complète sur ce système. On a étudié les résultats obtenu dans les différents algorithmes SVM, NB et KNN avec des différents représentations et en décide de prendre l'algorithme SVM comme meilleur classifieur grâce a ces bonne résultats.

CONCLUSION GENERALE

Le domaine de détection de spam a particulièrement progressé ces dix dernières années, grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré significativement le taux du filtrage de spam, par la progression de classification des emails en spam et légitime.

À l'heure actuelle, les techniques de filtrage de spam à base d'apprentissage sont loin d'être performants à 100%, car ces dernières ne traitent pas de la sémantique. Donc, il est très important de continuer à progresser d'une part dans le domaine de traitement linguistique de textes, afin d'arriver à une représentation textuelle manipulable par l'algorithme de classification utilisé en gardant la sémantique du texte. Et d'autre part, utiliser des algorithmes de classification performants pour la classification des courriels.

Récemment, les chercheurs ont examiné l'utilisation de la sémantique dans le filtrage de spam en représentant des emails avec un nouveau modèle vectoriel qui utilise une ontologie pour représenter les différentes relations entre les termes et, de cette manière, il offre un modèle plus riche. Sur la base de cette représentation, ils appliquent plusieurs classifieurs bien connus (SVM, NB, K-ppv et AD) et montrent que la méthode proposée permet de détecter la sémantique interne des messages et que cette approche donne des pourcentages élevés de détection de spam.

L'objectif de notre travail se dirigeait vers le développement d'une approche d'apprentissage automatique, afin d'améliorer la performance du système de filtrage avec SVM. Malgré les performances de ce système, il est intéressant de continuer le travail sur d'autres corpus, et appliquer une combinaison avec d'autres classifieurs tels que : naive bayes, les réseaux de neurones, etc.

Bibliographie

[1] Sanz E.P. et al, Email spam filtering, Advances in computers, vol.74, 2008, pp. 45-114.

[2] Guzella T.S., Caminhas W.M, A review of machine learning approaches to Spam filtering, Expert Systems with Applications, 2009, pp. 10206-10222.

[3] wikipedia, www.wikipedia.com, consulté le : 14/04/2018

[4] arobase, www.arobase.org, consulté le : 28/04/2018

[5] aidewindows, www.aidewindows.net/phishing.php, consulté le : 28/04/2018

[6] sebsauvage, www.sebsauvage.net/comprendre/spam/index.html, consulté le : 28/04/2018

[7] B. Hassan, Algorithme de boosting et méta-heuristique basée sur la PSO Pour La détection et le Filtrage De Spam, Thèse de Master, Université Tahar Moulay-SAIDA, 2013

[8] statista, www.statista.com, consulté le : 29/04/2018

[9] S. Gastellier-prevost, Le spam, 2009.

[10] G. Schryen, Anti-Spam Measures Analysis and Design, Berlin Heidel berg New York, Springer, 2010.

[11] anti-spam, www.anti-spam.fr, consulté le : 02/05/2018

[12] Nouman Azam, Comparative Study of Features Space Reduction Techniques for Spam Detection, Thèse de Master, National University of Sciences & Technology, Pakistan.

[13] frameip ,www.frameip.com/spam-ham-antispam, consulté le : 04/05/2018

[14] M. S. El Bazzi, T. Zaki, D. Mammass, A. Ennaji, Indexation automatique des textes arabes : état de l'art, 2016.

[15] B.S. Harish, D.S. Guru et S.Manjunath, Representation and Classification of Text Documents: A Brief Review, Recent Trends in Image Processing and Pattern Recognition, RTIPPR, 2010.

[16] D. Lewis, Feature Selection and Feature Extract ion for Text Categorization, pp. 212-217.

[17] Radwan JALAM, Apprentissage automatique et catégorisation de textes multilingues, Thèse de doctorat, Université Lumière Lyon 2, Juin 2003.

[18] N.Béchet, Extraction et regroupement de descripteurs morpho-syntaxiques pour des processus de Fouille de Textes, Thèse de doctorat, Université de MontPellier II, 2009.

[19]A. Cornuéjols, et L. Miclet, Apprentissage artificiel Concepts et algorithmes, Eyrolles, 2002.

[20] Sholom M. Weiss, Nitin Indurkhya, Tong Zhang, Fred J. Damerau, Text Mining Predictive Methods for Analyzing Unstructured Information, Springer-Verlag New York, 2005.

[21] Rocchio, J. J., The SMART Retrieval System: Experiments in Automatic Document Processing, Relevance Feedback In Information Retrieval, 1971, pp. 313-323.

[22] Bayes, T., An Essay towards solving a Problem in the Doctrine of Chances. Vol. 53, 1763, Philosophical Transactions of the Royal Society of London.

[23] N. H. Rimouche, H. Hachemi, Amélioration du produit scalaire via les mesures de similarités sémantiques dans le cadre de la catégorisation des textes, Thèse de Master, Université Abou Bakr Belkaid– Tlemcen, 2015.

[24] H. Drucker, Donghui Wu et Vladimir N. Vapnik, Support Vector Machines for Spam Categorization, IEEE transactions on neural networks, Vol. 10, 1999, pp. 1048-1054.

[25] Saumya Goyal et al, spam detection using KNN and Decision Tree mechanism in social network,Fourth International Conference on Parallel Distributed and Grid Computing, 2016, pp. 522-526.

[26] Nurul Fitriah Rusland et al, Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets, Conference Series: Materials Science and Engineering, 2017.

[27] Anju Radhakrishnan et V. Vaidhehi, Email Classification Using Machine Learning Algorithms, International Journal of Engineering and Technology,Vol. 9, 2017, pp. 335-340.

[28] Shradhanjali et Toran Verma, E-Mail Spam Detection and Classification Using SVM and Feature Extraction, International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 3, 2017, pp. 1491-1495.

[29] D. S. Jawale, A.G. Mahajan, K.R. Shinkar et V. Katdare, Hybrid spam detection using machine learning, International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 4, 2018, pp. 2828-2832.

[30] WEKA, www.cs.waikato.ac.nz/ml/weka/, consulté le : 12/05/2018

[31] Department of Telematics, www.dt.fee.unicamp.br/~tiago/smsspamcollection, consulté le : 15/05/2018

[32] Barigou Baya Naouel, Detection de courriels indésirables par apprentissage automatique, Thèse de Magister, Université d'Oran, 2012.

ملخص

يشكل البريد الإلكتروني خدمة فعالة لمستخدمي الانترنت، إنه وسيلة سريعة واقتصادية لتبادل المعلومات. ومع ذلك يجد المستخدمون أنفسهم مكتوفي الأيدي مع كميات من الرسائل غير المرغوب فيها والتي أصبحت تشكل مشكلة كبيرة لهؤلاء المستخدمين.

في إطار هذا العمل، يتم تصنيف رسائل البريد الإلكتروني باستخدام ثلاثة خوارزميات هامة وهي: SVM،Naïve Bayes وKNN ، يتم اختبار كفاءة هذه الخوارزميات مع طرق تمثيل مختلفة باستخدام مجموعة المعطيات smsSpamCollection. تظهر نتائج الاختبار أن خوارزمية SVM أفضل من خوارزميات NB و.KNN. ولذلك سيتم تصنيف البريد الإلكتروني المستقبل بواسطة هاته الخوارزمية.

ABSTRACT

The e-mail really makes service to users, it is a fast and economical way to exchange information. However, users find themselves quickly overwhelmed with amounts of unwanted messages called spam. Spam has quickly become a major problem on the Internet.

As part of our work, the classification of e-mails is carried out using three important machine learning algorithms: Support Vector Machine, Naïve Bayes and K nearest neighbors, the efficiency of these classifiers is tested with different representations using the smsSpamCollection dataset. The test results show that SVM is better than the NB and KNN algorithms.

Keywords : Spam, Algorithm Machine Learning, Support Vector Machine, Naïve Bayes, K Nearest Neighbors.

RESUME

Le courrier électronique rend vraiment service aux usagers, c'est un moyen rapide et économique pour échanger des informations. Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de messages indésirables appelé aussi spam. Le spam est rapidement devenu un problème majeur sur Internet.

Dans le cadre de notre travail, la classification des courriers électronique est effectuée à l'aide de trois algorithmes d'apprentissage automatique importants : Machine à Vecteurs de Support, Naïve Bayes et K voisins les plus proches, l'efficacité de ces classificateurs est testé avec des différents représentations on utilisant le corpus smsSpamCollection. Les résultats des tests montrent que SVM est plus performant par rapport aux algorithmes NB et KNN.

Mots-clés : Spam, Algorithme d'apprentissage automatique, Machine à Vecteurs de Support, Naïve Bayes, K voisins les plus proches.