Sélection de critères pour le filtrage automatique de messages

O. Nouali

(1) Laboratoire des Logiciels de base, CE.R.I.S.T,
Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie
Fax : 213 21 912126 - Tél. : 213 21 916211
(2) LPL- Université de Provence
29, Av. Robert Schuman, F-13621 Aix-en-Provence, France.
Fax: +33 (0).42.59.50.96- Tél.: +33 (0) 42.95.36.23
E-mail : onouali@mail.cerist.dz

Mots-clefs – Keywords

Filtrage d'information, e-mail, réseaux de neurones, apprentissage, *spam*.

Information filtering, e-mail, neural network, learning, e-mail filtering, *spam*.

Résumé - Abstract

La plupart des systèmes de filtrage du courrier électronique existants enregistrent des lacunes ou faiblesses sur l'efficacité du filtrage. Certains systèmes sont basés seulement sur le traitement de la partie structurée (un ensemble de règles sur l'entête du message), et d'autres sont basés sur un balayage superficiel de la partie texte du message (occurrence d'un ensemble de mots clés décrivant les intérêts de l'utilisateur).
Cet article propose une double amélioration de ces systèmes. D'une part, nous proposons un ensemble de critères automatisables et susceptibles d'influer sur le processus de filtrage. Ces critères sont des indices qui portent généralement sur la structure et le contenu des messages. D'autre part, nous utilisons une méthode d'apprentissage automatique permettant au système d'apprendre à partir de données et de s'adapter à la nature des mails dans le temps. Dans cet article, nous nous intéressons à un type de messages bien particulier, qui continue à polluer nos boîtes emails de façon croissante : les messages indésirables, appelés *spam*. Nous présentons à la fin les résultats d'une expérience d'évaluation.

Most of existing filtering messages systems exhibit weaknesses in term of efficiency. In fact, there are systems that use only message header information and others use a superficial processing of message body. In this paper, we try to improve the filtering processes efficiency. First, we introduce a set of criteria which are cues related to the message structure and content. Second, we use a machine learning method allowing the system to learn from data and to adapt to the email nature. We are interested in a special type of messages that continuously poluate our email boxes: *spam* email. At the end, to measure the approach performances, we illustrate and discuss the results obtained by experimental evaluations.

1 Introduction

Aujourd'hui, le courrier électronique est le mode de communication le plus populaire. Il est devenu un moyen rapide et économique pour échanger des informations. Cependant, les utilisateurs d'Internet se retrouvent assez vite submergés de quantités astronomiques de messages dont le traitement nécessite un temps considérable.
Dans cet article, nous nous intéressons à un type de messages bien particulier, qui continue à polluer nos boîtes emails de façon croissante : les messages indésirables, appelés *spam*. Par exemple, messages proposant des services, des produits miraculeux (maigrir en un temps record, etc.), offres de voyages à prix attractif, opportunités d'investissement pour devenir riche en peu de temps, propositions de cartes de crédit à taux d'intérêt réduit, messages pornographiques, etc. Le *spam* est un phénomène mondial et massif. Il cause de multiples désagréments tels que l'engorgement des boîtes emails et des serveurs emails, dilution des messages utiles, perte de temps et d'espace, etc. Certains systèmes de filtrage de *spam* existants sont basés seulement sur le filtrage des adresses émettrices en se basant sur une liste noire des *spammeurs*, et d'autres permettent aux utilisateurs d'écrire manuellement de règles logiques de filtrage à base de mots clés. Le problème avec ces systèmes, est que d'une part, ils sont moins précis et d'autre part la nature des messages *spam* varie au cours du temps, ce qui nécessite une mise à jour fréquente de ces règles. Pour palier ce problème, une solution est de développer des systèmes évolutifs qui s'adaptent à la nature des mails au cours du temps et donc utiliser les techniques d'apprentissage automatique à partir de données. De nombreux travaux, dans le domaine d'apprentissage, ont porté sur la classification de textes (Yang, Pedersen, 1997; Sebastiani, 1999) et peu de travaux ont porté sur le filtrage de *spam* (Sahami et al., 1998; Orasan, Krishnamurthy, 2002). Une grande majorité de ces travaux utilise la cooccurrence lexicale comme base de leur classification.
Dans cet article, nous proposons une solution évolutive qui s'adapte à la nature des mails dans le temps et permet un filtrage nettement meilleur en qualité, basé sur un ensemble d'indices portant généralement sur la structure et le contenu des messages.

2 Les critères de filtrage

Nous avons défini et identifié un ensemble de critères que nous avons classés en trois types (table 1).

Mots simples (MS)
business, time, money, free, price, product, credit, opportunity, guarantee, marketing, investment, risk, advertisement, sex, travel, miracle, etc.
Mots composés ou phrases très courtes
business opportunity, credit card, free investment, half price, home business, immediate release, investment report, limited time, special bonus, take action, etc.
Caractéristiques spécifiques
le domaine des adresses émettrices, la longueur de l'entête, le type du message, abréviations, les caractères non alphanumérique, les caractères numériques, la langue, les fichiers attachés, horaire d'envoi, etc.

Table 1 : Principaux critères de filtrage

Les mots simples représentent le vocabulaire de base, généré automatiquement à l'aide de la mesure de l'information mutuelle (Yan, 1997). Les mots composés sont générés à partir des listes *bigrammes* et *trigrammes* apprises par le système. Les caractéristiques spécifiques représentent l'ensemble d'indices portant sur la structure et le contenu des messages.

Voici quelques résultats de l'étude de notre corpus:

- Le domaine des adresses émettrices (.com, .gov, .edu, etc.) : 52% de *spam* contre 6% de *non spam* pour le domaine *com,* 13% contre aucun pour le domaine *net,* etc.
- La longueur de l'entête des messages: les messages *spam* subissent avant d'être reçus par le destinataire, un certain nombre de relais par des serveurs de mails de façon à atteindre un maximum d'utilisateurs. 96 % de *spam* de la base subissent des relais contre aucun pour *non spam*[1].
- Le type du message : 45% de *spam* sont de type *html*, contre aucun pour *non spam*.
- La longueur du message : elle est évaluée par le nombre de mots. 85% de *spam* sont de taille relativement courte contre 95% pour *non spam*.
- Mots non fréquents (noms communs : catégorie non reconnue par l'analyseur) : Ce sont tous les mots qui commencent par une majuscule et les mots non étiquetés par l'analyseur. 86% de *spam* contre 5% pour *non spam*.
- Abréviations : 35% de *spam* contre 10% pour *non spam.*
- Les caractères non alphanumérique ($, !, #, %, *, &, etc.): 65% de *spam* contiennent des caractères non alphanumériques contre 2% pour *non spam*. 76 % de *spam* contiennent le point d'exclamation (ex : *get rich quick!*) contre 90% de *non spam* ne le contiennent pas. 43 % de *spam* contiennent le caractère $ (90% dans le champ *subject*) contre aucun pour *non spam*.
- Les caractères numériques : 80% de *spam* contre 10% pour *non spam*.
- La langue du message : 100% de *spam* sont en anglais contre 80% de *non spam* sont en français.
- Les fichiers attachés : 98% de *spam* n'ont pas de fichiers attachés contre 92% pour *non spam*.
- La taille des phrases : 76% de *spam* contiennent des phrases courtes (<10 mots) contre 92 % pour *non spam*.
- Horaire d'envoi (nuit/jour) : 65% de *spam* sont envoyés la nuit contre 88% de *non spam,* sont envoyés le jour.
- etc.

3 Le modèle de connaissance

Il représente l'ensemble des caractéristiques du domaine *spam* définies et identifiés par apprentissage à partir d'un corpus de messages. Le modèle adopté pour modéliser le profil *spam* est un réseau de neurones non récurrents (absence de boucles) à trois couches (figure 1). Une couche en entrée qui reçoit les entrées du réseau. Une couche cachée représentant l'ensemble des caractéristiques du profil *spam*. Une couche de sortie qui représente deux types de messages : *spam* et *non spam*.
Dans un réseau de neurones, la connaissance est codée par la valeur des poids des différentes connexions. Ce codage est estimé par apprentissage. Il s'agit, à partir d'un ensemble d'exemples observés, d'estimer les paramètres du réseau de neurones. Le réseau est entraîné par l'algorithme de propagation arrière ou *rétro-propagation* qui consiste à corriger les poids des connexions des différentes couches en fonction des erreurs commises. La correction se fait de la couche de sortie à la couche d'entrée. La fonction d'activation choisie pour toutes les couches est la fonction sigmoïde (Davalo, Naim, 1993).

[1] Un relais est identifié dans l'entête du message par le mot clé : « Received : from… »

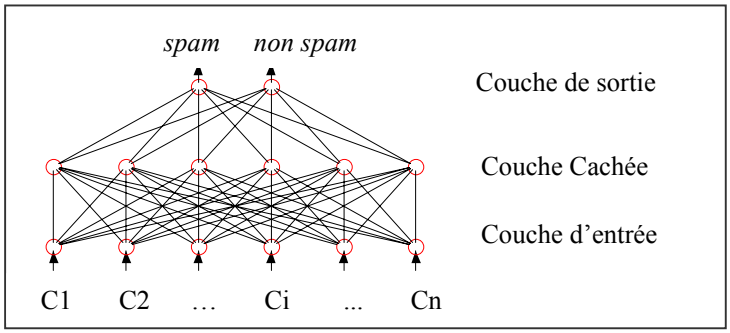

Figure 1 : Architecture d'un réseau à trois couches

L'algorithme d'apprentissage est décrit brièvement comme suit:

(1) Etiqueter chaque message du corpus (*Spam* ou *non spam*).
(2) Faire passer le corpus par les différents modules d'analyse pour avoir la représentation associée de chaque vecteur.
(3) Initialiser les paramètres du réseau : au départ les poids des connexions entre neurones des différentes couches sont définis par défaut à 0.5, et l*e pas d'apprentissage* initialisé à *0.1*.
(4) Lancer l'apprentissage qui consiste à : calculer la sortie du réseau pour chaque message, comparer et calculer l'erreur, et mettre à jour les paramètres du réseau (ajuster les poids).

4 Le processus de filtrage

En premier, un module de ***pré-traitement*** est lancé pour préparer les messages, récupérés de la boîte email, aux différentes étapes ultérieures de l'analyse. Il consiste à isoler les différents champs et à identifier la langue de chaque message parmi deux actuellement modélisées (Français, Anglais). Par ailleurs, le système est incrémental et permet facilement la prise en compte de nouvelles langues (ajouter un anti-dictionnaire propre à chaque nouvelle langue).

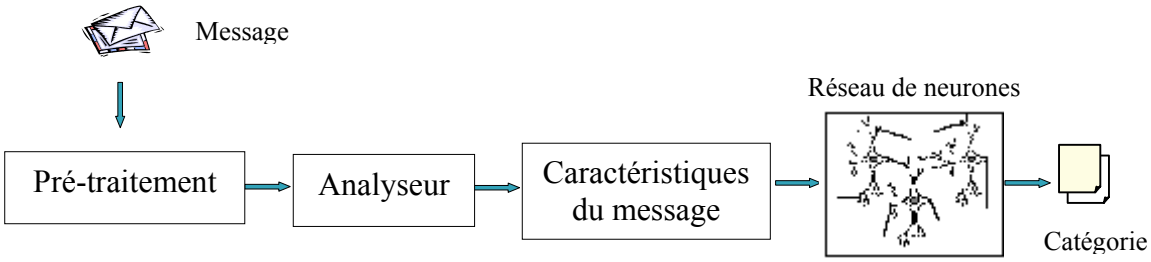

Figure 2 : le processus de filtrage

Après l'étape de ***pré-traitement***, le message est donné à un analyseur, qui a pour but d'identifier les informations pertinentes à représenter. Il s'agit d'extraire les différentes propriétés permettant de le caractériser et de construire en sortie le vecteur message associé. En effet, le message est représenté conceptuellement par un espace vectoriel de k dimensions: $M = \{(T1,W1), (T2,W2, \ldots , (Tk,Wk)\}$ (Ti: ième caractéristique, Wi: poids et k : espace des caractéristiques). Cette représentation constitue l'entrée du réseau de neurones. Elle est donc crée dynamiquement à chaque récupération d'un nouveau message. Ce vecteur sera propagé à travers les différentes couches du réseau pour donner en sortie le type du message : *spam* ou *non spam*. L'identification de certains critères nécessite une phase d'étiquetage préalable. En effet, nous avons utilisé un étiqueteur morpho-syntaxique, analyseur de Brill (Brill, 1992).

Le système dispose d'un **apprentissage assisté** appelé ***feed-back*** *(Oubbad, Nouali, 1999)* où l'utilisateur est invité à donner son avis sur le comportement du système, ce qui lui permet d'approcher la pertinence de l'utilisateur et de s'adapter ainsi à ses besoins.

5 Evaluation

Pour effectuer nos tests nous avons travaillé avec un corpus de 1000 messages construit à partir d'un ensemble de messages que nous avons collectés pendant quatre mois, contenant 700 mails de classe *spam* et 300 *non spam*. Nous avons divisé le corpus en une base d'apprentissage et une base de tests selon le découpage suivant :

- Base d'apprentissage : 450 *spam* et 200 *non spam*.
- Base de test : 250 *spam* et 100 *non spam*.

Pour mesurer les performances nous avons utilisé les mesures suivantes :

$$Rappel = \frac{\alpha}{\alpha + \gamma} \qquad Précision = \frac{\alpha}{\alpha + \beta}$$

$$Erreur\_globale = \frac{\beta + \gamma}{\alpha + \beta + \gamma + \delta} \qquad Précision\_globale = \frac{\alpha + \delta}{\alpha + \beta + \gamma + \delta}$$

avec : α: Messages *spam*, correctement filtrés (classés) par le système.
β : Messages *non spam*, incorrectement filtrés par le système.
γ : Messages *spam*, incorrectement non filtrés (rejetés) par le système.
δ : messages *non spam*, correctement non filtrés par le système.

Expérience 1: *Performances en fonction des caractéristiques considérées*
Nous mesurons les performances du système en considérant tout d'abord un modèle de base constitué uniquement de mots simples et lorsque nous ajoutons des critères supplémentaires.

Caractéristiques	***Spam***		***Non spam***		**Performance globale**	
	Précision	***Rappel***	***Précision***	***Rappel***	***Erreur Globale***	***Précision Globale***
MS	97,8%	81,8%	77,2%	97,1%	12,2%	87,7%
MS + MC + CS	99,4%	86,3%	82,2%	99,2%	8,6%	91,4%
+Pondération	99,5%	**95,4%**	**93,2%**	99,2%	**3%**	**97%**

Tableau 1 : les performances du système

Au début de l'expérience, on remarque un taux d'erreur de 12%, et un écart significatif entre le taux de *précision* et le taux de *rappel.* En effet, le système considère certains *spam* comme des messages légitimes. Nous reprendrons les tests en ajoutant les critères définis précédemment. Initialement, nous avons attribué un même poids à tous les critères de filtrage. Nous constatons une amélioration des performances mais non maximale. Ensuite, nous avons modifié l'importance des différents critères, en attribuant une forte valeur du poids à certains critères et aux termes spécifiques, qui sont uniques dans le type *spam*. Les résultats des tests étaient beaucoup meilleurs (95% de *rappel*).

Expérience 2: *mesurer l'importance et le rôle de l'apprentissage*
L'expérience consiste à présenter au système en deux cas différents, un ensemble de courriers à filtrer en plusieurs sessions. Puis mesurer à chaque fois le taux global de succès du système et effectuer un apprentissage assisté pour mesurer son efficacité et son influence sur les deux facteurs (figure 3). Nous constatons que le modèle nécessite plusieurs sessions d'apprentissages assistés pour améliorer la qualité de ses résultats. Il est donc nécessaire de lancer l'apprentissage feedback régulièrement, par exemple après chaque session de filtrage.

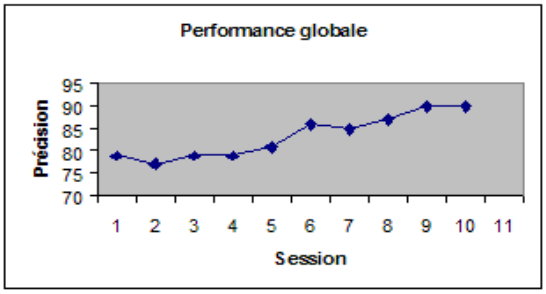

Figure 3: l'importance de l'apprentissage assisté

6 Conclusion

Cet article propose une solution évolutive qui s'adapte à la nature des mails dans le temps et permet un filtrage beaucoup meilleur en qualité. En effet, un traitement statistique de corpus nous permet de proposer un certain nombre de critères permettant d'améliorer les résultats de filtrage. Ces critères sont modélisés par un réseau de neurone. Chaque nœud du réseau correspond à un critère, qui est pondéré par un poids qui représente son degré d'importance. Un module d'apprentissage permet d'améliorer les résultats du système. Les résultats de tests obtenus sur notre corpus semblent satisfaisants. Néanmoins, pour tester l'adaptabilité de l'approche, il serait intéressant d'étendre les tests sur un échantillon de messages plus important.

Références

Brill E., (1992), A Simple Rule-based Part of Speech Tagger, *Proceedings of the Third Conference on Applied Natural Language Processing, ACL,* pp.152-155, 1992.

Cohen W. W., (1996), Learning rules that classify e-mail, *In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access,* 1996.

Davalo E., Naim P., (1993), Des Réseaux de Neurones, *Edition Eyrolles,* 1993.

Nouali O., (2002), Classification Automatique de messages : une approche hybride, *RECITAL2002*, Nancy, 24-27 juin 2002.

Oubbad L., Nouali O., (1999), Système de filtrage du courrier électronique, *Mémoire d'ingénieurs*, INI, Alger, novembre 1999.

Orasan C., Krishnamurthy R., (2002), A corpus-based investigation of junk emails, *In Proceedings of LREC-2002,* Las Palmas, Spain, 2002.

Sahami M., Dumais S., Heckerman D., Horvitz E, (1998), A Bayesian approach to filtering junk e-mail, *In Learning for Text Categorization Papers from the AAAI Workshop,* pp. 55–62, Madison Wisconsin, AAAI Technical Report WS-98-05, 1998.

Sebastiani F., (1999), A Tutorial on Automated Text Categorisation, *Proceedings of ASAI-99, 1st Argentinean Symposium on Artificial Intelligence,* 1999.

Yang Y., Pedersen J. O., (1997), A comparative Study on Feature Selection in Text Categorization, *International Conference on Machine Learning, ICML*, Nashville, TN, USA, 1997.