Méthodologie d'évaluation des filtres anti-spam

José-Marcio Martins da Cruz
Mines ParisTech – Centre de Calcul et Systèmes d'information
60, bd Saint Michel
75006 - Paris

Résumé

Depuis une dizaine d'années, les administrateurs de messagerie utilisent des outils divers, généralement des filtres distribués sous licence libre, pour lutter contre le spam. On les met en place, assez souvent, grâce à des arguments qui nous semblent convaincants, ou par indications faites par des collègues, mais leur efficacité est rarement connue ou évaluée sérieusement. Le marché de la sécurité informatique, étant un marché très concurrentiel, les fournisseurs ne renseignent pas toujours les vrais indices d'efficacité de leurs produits.

Cet article présente une méthodologie d'évaluation de filtres anti-spam, prioritairement adaptée pour l'évaluation de filtres de contenu, mais qui peut être transposée pour les filtres protocolaires. Nous passons en revue les particularités du filtrage de spam, les points importants à prendre en compte dans une évaluation ainsi que les indicateurs d'efficacité de classement usuels et pertinents pour le cas de filtrage de spam.

Mots clefs

Filtrage de spam, évaluation, filtrage en ligne

1 Introduction

Dans la deuxième moitié des années 90, la problématique du filtrage de spam par contenu, a attiré l'attention de trois domaines de recherche qui n 'étaient pas directement concernés par la messagerie électronique : la Recherche Documentaire (IR), la Fouille de Données (DM) et l'Apprentissage Automatique (ML)[1]. Le premier résultat de recherche publié d'utilisation d'un filtre statistique pour filtrer le spam date de 1998, par Sahami et al [1], utilisant un classificateur bayésien naïf. En 2002, Paul Graham a publié un billet sur son blog[2] qui a déclenché un foisonnement de filtres statistiques, dits bayésiens, distribués sous licence libre. Plusieurs centaines de communications scientifiques ont été publiées depuis, et de nombreux logiciels de filtrage sont disponibles (commerciaux ou libres).

On est souvent confronté à des affirmations du genre : « ce filtre anti-spam est capable de détecter 99 % des spams sans aucun faux positif ». Cette information semble intéressante mais, sauf à avoir plus de précisions, elle n'est pas vraiment utile. Quel administrateur de messagerie ne s'est jamais posé la question : quelle est l'efficacité de mon filtre anti-spam ? Si j'essaye telle ou telle modification dans la configuration de mon filtre, comment vérifier si cela en améliore l'efficacité ? Comment comparer deux filtres anti-spam ?

Les chercheurs et les développeurs ont besoin d'évaluer les filtres pour mettre au point et sélectionner des méthodes de filtrage. Les gestionnaires de services de messagerie ont le même besoin d'abord pour décider lequel convient le mieux à leur organisation et ensuite pendant la phase de production, pour détecter toute déviation par rapport à l'efficacité initiale.

Le marché de la sécurité informatique, en général, et de la messagerie en particulier est très concurrentiel et on ne doute pas que les fournisseurs soient réticents à ce que leurs produits puissent être évalués et comparés avec d'autres.

Vaderetro indique, dans la plaquette de présentation de *Mail-Cube*[3], une « d*étection importante des spams (95 %)* » mais ne précise ni les conditions d'évaluation ni le taux de faux positifs. Il n'y a pas plus d'information dans le manuel utilisateur. De même, il est indiqué que *« plus de 100 messages à la seconde peuvent être analysés sur une machine équipée d'un Pentium 4 cadencé à 1,9 GHz »*, mais aucune précision n'est donnée sur la taille des messages, ni s'il s'agit de messages refusés par réputation du client (auquel cas il vaut mieux parler de connexions et non pas de messages).

La plaquette commerciale de *Ironport*[4], se limite à dire que le boîtier est capable de *« refuser jusqu'à 80 % du spam dans la phase de connexion, sans aucun risque de faux positifs »*. Il n'est pas fait mention de l'efficacité du filtrage de contenu. Dans les

[1] Ces domaines sont connus dans la littérature en langue anglaise par les noms « Information Retrieval », « Data Mining » et « Machine Learning ».

[2] P. Graham. A plan for spam. http://www.paulgraham.com/spam.html, 2002.

[3] Vaderetro – Documentation commerciale : http://www.vade-retro.com/doc/plaquette_vaderetro.pdf

[4] Ironport – Documentation (Datasheet) http://www.ironport.com/pdf/ironport_anti-spam_datasheet.pdf

messages traités par le filtre de contenu, des entêtes sont ajoutés, mais les informations sont codées ce qui rend difficile (mais pas impossible) toute évaluation faite par des tiers, utilisant les valeurs de score.

Lors de la « Spam Conference » du MIT en 2004, Bill Yerazunis a fait une présentation avec le titre « *The Spam-Filtering Accuracy Plateau at 99.9 % Accuracy and How to Get Past It* » [2]. Un an après, lors de la conférence TREC – Spam Track 2005 [3], quatre configurations de son filtre ont été testées contre quatre corpus de messages : les résultats d'exactitude ont varié entre 87,95 % et 99,66 % et n'ont donc pas atteint le niveau annoncé par l'auteur. L'utilisation d'exemples d'apprentissage dans l'ensemble à tester et la simulation effectuée sans tenir compte de l'ordre chronologique des messages ne seraient valables que si le spam était un processus stationnaire : ce sont les erreurs commises par Yerazunis.

La différence entre les mesures effectuées peut même donner naissance à des situations conflictuelles. Gordon Cormack a comparé [4] l'efficacité de 6 filtres anti-spam open source (Bogofilter, SpamAssassin, Dspam, CRM114, SpamProbe et SpamBayes). Jonathan Zdziarski, auteur de Dspam, mécontent de voir son filtre classé en avant-dernière place (derrière SpamAssassin et Bogofilter) a initié une polémique. Pourtant, le protocole d'évaluation utilisé par Gordon Cormack est largement plébiscité par la communauté de la recherche, alors que Jonathan Zdziarski n'évalue [5] que l'exactitude (*accuracy*), paramètre qui n'est pas pertinent pour une application de classement tel le filtrage de spam [6], avec un protocole d'évaluation moins justifié que celui de Gordon Cormack.

Les caractéristiques intrinsèques des filtres (par exemple, le type de classificateur et les caractéristiques d'extraction des attributs) font que certains filtres donnent de meilleurs résultats que d'autres. Ajoutons que les facteurs externes (par exemple, les jeux de données, le protocole d'apprentissage, l'environnement linguistique) font que le classement d'un filtre peut varier d'une évaluation à une autre.

Il est donc nécessaire non seulement d'évaluer les filtres dans des conditions les plus proches possibles des conditions réelles d'utilisation, mais aussi de les préciser. Il est également parfois intéressant d'évaluer la sensibilité du filtre à des changements limités des conditions d'évaluation avec, par exemple, l'ajout de bruit.

Le but de ce papier est de présenter les difficultés de l'évaluation des filtres anti-spam, les mesures pertinentes et les idées de méthodologie utilisées dans TREC Spam Track, méthodologie qui est actuellement la plus acceptée par la communauté de la recherche.

Il y a deux catégories de filtres anti-spam, avec des frontières parfois floues :

- **filtres protocolaires ou comportementaux** : ce sont des filtres dont le fonctionnement est basé sur des paramètres liés au protocole SMTP tels les listes noires ou de réputation, les cadences de connexion et le respect du protocole SMTP ;
- **filtres de contenu** : ce sont les filtres qui ne tiennent compte que du contenu de la partie DATA du dialogue SMTP, c'est-à-dire tout ce que l'on peut déduire des entêtes et du corps du message.

L'évaluation des filtres protocolaires est souvent plus délicate puisqu'elle dépend d'événements qui ne sont présents que dans des conditions réelles de fonctionnement ou alors de paramètres qui ne sont pas toujours observables tels le classement des messages refusés par une liste de réputation d'adresses : dans ce cas précis, on ne peut pas évaluer le taux de faux positifs. L'évaluation de ces filtres dépend souvent de la mise en place d'un dispositif spécifique au type de filtrage. Dans le cas d'une liste de réputation, par exemple, cela consiste à recevoir tous les messages, même ceux qui seraient refusés à tort ou à raison, en y ajoutant une information de marquage pertinente. À partir du moment où l'évaluation de ces paramètres est possible, les principes sont les mêmes que pour l'évaluation des filtres de contenu. Certains aspects de l'évaluation de ces filtres sont cités dans cet article, mais nous nous attachons surtout à l'évaluation des filtres de contenu.

2 TREC – Spam Track

TREC (« Text REtrieval Conference »)[5] est un workshop sponsorisé par deux départements de l'administration américaine : le NIST (National Institute of Standards and Technology) et le DoD (Department of Defense). Le but est de promouvoir des échanges annuels entre chercheurs de tous horizons, autour de thèmes novateurs en rapport avec la problématique de la recherche documentaire.

Le thème Spam a été traité pendant trois ans, de 2005 à 2007. Le but était simple : évaluer si les filtres anti-spam étaient vraiment efficaces. À l'époque, les produits commerciaux étaient proposés avec une « sauce secrète » sans aucune possibilité de réelle vérification de leurs caractéristiques (et c'est toujours le cas). Les filtres anti-spam libres évoquaient une excellente efficacité, mais avec des mesures empiriques on non reproductibles [2] [4]. La communauté de la recherche en intelligence artificielle étudiait le filtrage de spam en utilisant un modèle abstrait d'apprentissage supervisé avec des vecteurs d'attributs déjà extraits des messages, comme si l'algorithme de classement était la partie la plus critique du filtrage anti-spam.

[5] NIST Text REtrieval Conference - TREC, http://trec.nist.gov

Le principe de TREC a toujours été d'améliorer l'état de l'art grâce à des évaluations, mesures et comparaisons effectuées conjointement à l'aide de jeux de données publiquement disponibles. La problématique du spam est différente dans le sens où il s'agit d'un problème de classement en ligne, sur un flot de données non stationnaire et où pratiquement tous les jeux de données sont privés.

Gordon Cormack et Thomas Lynam[6] ont développé une boîte à outils avec une interface permettant de soumettre au filtre, pour classement, une suite de messages selon un ordre chronologique précis et d'enregistrer les réponses. Les indicateurs d'efficacité pertinents ainsi que leur signification statistique sont évalués à partir des réponses du filtre. Deux corpus de messages ont été utilisés : l'un public constitué à partir des messages récupérés après la faillite de la société Enron et l'autre privé constitué à partir des messages reçus par un utilisateur sur une période de 8 mois.

L'efficacité des filtres participants à TREC Spam Track s'est améliorée chaque année, par rapport à l'année précédente. En 2005, Andrej Bratko et Bogdan Filipic [7] ont démontré que les modèles à compression étaient plus performants que les filtres pseudo-bayésiens de l'époque. En 2006, le filtre proposé par Fidelis Assis [8], utilisant des digrammes orthogonaux épars pour modéliser les messages, avec un seuil d'apprentissage, a été le plus performant. En 2007, les filtres basés sur des classificateurs à régression logistique [9] et SVMs (Support Vector Machines) [10] et utilisant des n-grams au lieu de mots pour modéliser les messages ont été les plus performants.

Le résultat le plus intéressant de TREC a été la mise au point et validation d'une méthodologie d'évaluation de filtres anti-spam.

3 Évaluation de filtres anti-spam – principes et challenges

Le processus de filtrage de spam est généralement représenté par le modèle de la Figure 1. Une suite de messages est présentée, de façon séquentielle, à un classificateur qui les traite également de façon séquentielle et les attribue à une classe (ham ou spam). Le destinataire peut garder ou détruire le message, selon le classement donné par le filtre. Il peut aussi vérifier si le classement du filtre est correct et retourner (ou pas) des informations d'erreur, de façon à mettre à jour les modèles sur lesquels le filtre base son jugement (c'est l'*apprentissage en ligne*). Il convient de considérer que le retour d'information de la part du destinataire n'est ni systématique ni immédiat et, de plus, peut être erroné.

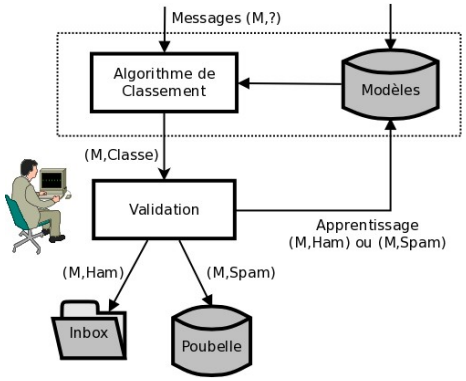

Figure 1: Modèle Générique d'un processus de filtrage de spam

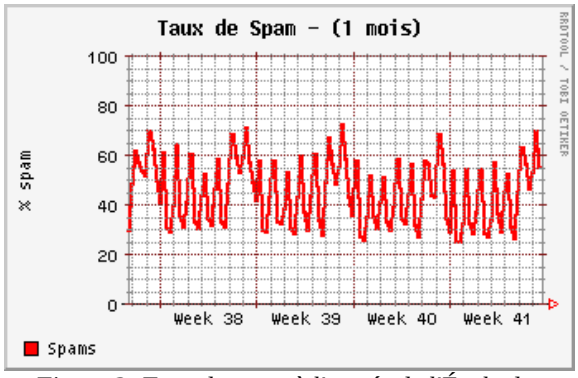

Figure 2: Taux de spam à l'entrée de l'École des Mines de Paris après greylisting

3.1 Les exigences des méthodes d'évaluation

Le protocole d'évaluation d'une « boîte noire » de classement, que ce soit un filtre anti-spam ou non, doit satisfaire quelques contraintes :

- lorsque l'évaluation est faite hors ligne, elle doit reproduire aussi fidèlement que possible l'environnement et l'utilisation réelle du filtre. Une erreur souvent commise est de considérer que le spam est un processus qui n'évolue pas dans le temps. Dans un autre exemple, un filtre de contenu placé après un filtre de *greylisting* n'est pas soumis au même type de contenu, ni en quantité, ni en qualité, que s'il n'y avait pas de filtrage par *greylisting ;*
- le test doit évaluer des critères d'efficacité significatifs et pertinents. Parmi la multitude de critères utilisés dans les différentes applications de classement statistique (diagnostic médical, classement de textes, recherche documentaire...), tous ne sont pas significatifs pour le classement du spam ;
- le test doit être effectué dans un environnement maîtrisé et stable. Ceci ne veut pas dire que l'on ne puisse pas tenir compte des événements inattendus, à condition qu'ils soient quantifiables en valeur, datés et que leur impact sur la mesure soit aussi quantifiable et limité ;

[6] TREC spam filter evaluation toolkit. http://plg.uwaterloo.ca/~gvcormac/jig/

– les résultats de l'évaluation doivent être statistiquement valides. Il faut non seulement pouvoir évaluer des valeurs mais également la confiance que l'on peut leur accorder, comme par exemple l'estimation de l'erreur de la mesure ;
– le test doit être reproductible. Comme toute expérimentation scientifique, tout doit être mis en œuvre pour que les résultats soient vérifiables. Cela implique que toutes les conditions de test doivent être datées et enregistrées. Sauf si l'évaluation n'a qu'un intérêt interne, les ensembles de données utilisés pendant l'expérimentation doivent être accessibles, ou alors assimilables (de façon vérifiable) à des données publiques.

3.2 Les aspects à prendre en compte lors de l'évaluation

Le filtrage de spam a des caractéristiques particulières qui doivent être prises en compte lors d'une évaluation [11] :

3.2.1 Non-stationnarité

Il s'agit d'un aspect particulièrement important du filtrage de spam et rarement pris en compte dans les procédures d'évaluation. Le trafic de la messagerie électronique varie selon trois aspects :

– la répartition des classes. La Figure 2 montre la variation du taux de spam à l'entrée de l'École des Mines de Paris. À noter qu'il s'agit d'une mesure après filtrage par le *greylisting* (la dynamique serait plus importante si la mesure avait été faite avant). On peut distinguer l'activité de nuit et de weekend. Cet exemple montre une variation à court terme, mais on remarque sur une période plus longue (une année, par exemple) une variation plus ou moins périodique dans les hams, en apparence liée aux vacances et fêtes et une variation non périodique des spams, plutôt liée à d'autres événements (tels le début ou la fin d'activité d'un spammeur, ou un événement lié à une célébrité) ;
– la répartition à l'intérieur de chaque classe. Ceci est plutôt vrai pour les spams. Rien ne garantit que la répartition par genre (pornographie, arnaques, médicaments...) soit constante dans le temps ;
– une variation qualitative à l'intérieur des messages. À l'intérieur des hams, les gens changent peu souvent leur façon d'écrire : les variations sont lentes. À l'intérieur des spams, au contraire, pour déjouer les filtres, les spammeurs changent souvent le contenu et la présentation des messages.

Ces trois aspects font que le spam n'est pas un processus statique (stationnaire). Les deux derniers aspects font que le respect de l'ordre chronologique des messages est important et la raison qui justifie que le filtrage de spam soit considéré comme un processus en ligne [12], même lorsque l'évaluation se fait hors ligne.

Des méthodes courantes telles que la validation croisée [13], plus adaptée aux processus stationnaires, ne peuvent donc pas être utilisées. Dans cette méthode, l'ensemble d'exemples est divisé en K sous-ensembles et l'on utilise à tour de rôle, K-1 sous-ensembles pour l'apprentissage et le dernier sous-ensemble pour l'évaluation.

3.2.2 Évaluation en ligne et aspects temps réel

Lorsqu'il s'agit d'un système en production, il est souvent souhaitable de pouvoir évaluer en ligne le filtre anti-spam. Mais, malgré le réalisme des conditions d'évaluation, certaines difficultés sont incontournables et relèvent, pour la plupart, du manque de maîtrise de l'environnement – les conditions de fonctionnement sont imposées par l'environnement avec ses aléas et l'évaluation de tous les paramètres de fonctionnement peut ne pas être possible. Cela fait qu'une évaluation n'est valable qu'au moment où elle est effectuée et elle est donc difficilement reproductible. Ainsi, ces évaluations sont plus intéressantes pour apprécier la stabilité dans le temps d'un filtre utilisé en production que pour évaluer son efficacité réelle.

Des événements externes imprévus peuvent modifier l'environnement et impacter l'efficacité de filtrage. Il faut être capable de les détecter et les enregistrer de façon à pouvoir expliquer les résultats inattendus. Parfois, ces enregistrements peuvent permettre de rejouer l'évaluation hors ligne. Certains changements dans l'environnement restent souhaitables et normaux (par exemple, la mise à jour d'une liste noire), mais rendent la mesure difficilement reproductible. Par contre, un événement tel l'inaccessibilité d'un serveur DNS d'une liste noire peut ne pas être détectable et fausser la mesure.

Il est aussi possible que certaines caractéristiques des filtres ne puissent pas être évaluées. Un exemple est le taux de faux positifs d'une liste noire, parce que des messages sont rejetés à la connexion et que la classe à attribuer ne peut pas être déterminée.

Si le but est l'évaluation intrinsèque de filtrage, une alternative souvent acceptable est d'enregistrer tous les événements survenus pendant une période donnée, d'accepter tous les messages (même ceux qui seraient refusés par une liste noire) et puis de les dérouler dans un environnement contrôlé.

Un challenge d'évaluation en ligne de filtres anti-spam a été proposé lors des éditions 2007 et 2008 de CEAS[7] avec des résultats intéressants, mais encore non satisfaisants. Le challenge de filtrage en ligne devra reprendre en 2010. Certains organismes font l'évaluation en ligne de filtres anti-spam[8]. Ce sont des évaluations intéressantes, mais vu qu'elles sont faites avec du trafic réel,

[7] The CEAS 2008 Live Spam Challenge. http ://www.ceas.cc/2008/challenge/challenge.html, 2008

elles ne sont valables qu'au moment où elles ont été réalisées : on ne peut donc pas comparer deux filtres qui n'ont pas été évalués en même temps et avec les mêmes messages à filtrer.

3.2.3 Interactions avec le destinataire

Le retour d'information sur le classement (correct ou non) est l'interaction la plus intéressante à étudier, puisqu'elle concerne directement les modes d'apprentissage du classificateur. Les résultats de filtrage influent et modifient le comportement de l'utilisateur et vice versa. La connaissance de cette interaction est un sujet encore ouvert et, de ce fait, on privilégie les évaluations hors ligne pour étudier ces interactions. Des exemples de scénarios étudiés sont :

- retour immédiat : c'est le cas le plus simple de l'utilisateur idéal qui retourne l'information d'exactitude de classement du message immédiatement après que le filtre l'a classé. Cette information est aussitôt intégrée à l'apprentissage du filtre, avant même l'arrivée du message suivant ;
- retour différé : c'est le cas de l'utilisateur qui ne lit son courrier qu'occasionnellement ou à des intervalles fixes : le retour d'information se fait quelque temps après le filtrage et provoque une baisse d'efficacité (à évaluer) ;
- retour sélectif : c'est le cas de l'utilisateur qui ne retourne l'information d'exactitude de classement que pour une partie des messages (par exemple, seulement les spams non détectés ou uniquement lorsqu'il a du temps libre) ;
- uniquement les erreurs : l'utilisateur ne retourne que les classements en erreur. Éventuellement, il peut ne s'agir que des erreurs sur une seule classe. Les utilisateurs regardent plus souvent les dossiers de messages légitimes que celui de spams et, de ce fait, remarquent plutôt les spams non détectés que les messages légitimes filtrés à tort ;
- retour sur demande : (aussi appelé Apprentissage Actif), il s'agit de la possibilité pour le classificateur de demander au destinataire le classement correct des messages dont le score se trouve dans une zone d'indécision ;
- erreurs de l'information de retour : le destinataire n'étant pas un classificateur parfait, les informations de classement qu'il retourne peuvent ne pas être exactes. Assez souvent, ceci est modélisé comme du bruit qui contribue à diminuer l'efficacité des filtres.

3.2.4 Interactions avec l'expéditeur

Les interactions du type *greylisting*, challenge/réponse, *captchas*, sont assez difficiles à simuler et à évaluer. Dans le cas du *greylisting*, par exemple, on ne peut pas évaluer dans un intervalle donné, la proportion de messages qui (avec le bon classement) ont été refusés grâce au *greylisting*, et le délai réel associé ne peut pas être évalué pour tous les messages acceptés. Ce qui se fait habituellement est de comparer les résultats d'évaluation faites avec et sans la fonctionnalité en question, mais la validité de la démarche est relative puisque le trafic de messagerie n'étant pas un processus stationnaire, on ne peut garantir que les deux évaluations ont été faites dans les mêmes conditions.

3.2.5 Corpus de messages utilisé pour l'évaluation

Des données différentes impliquent des résultats différents : rien ne peut assurer qu'un même filtre aura la même efficacité pour deux destinataires différents, ou pour le même destinataire à des instants différents. Il est donc important que des comparaisons soient faites en ayant comme référence le même ensemble de messages utilisé pour l'apprentissage et pour le test. S'il s'agit de comparer ses résultats avec des résultats obtenus par d'autres, il est essentiel que le corpus de messages soit le même.

La constitution d'un corpus public de messages n'est pas une tâche simple : il est difficile de convaincre quelqu'un de publier la totalité de ses messages. Les premières recherches ont été effectuées avec des corpus de messages en provenance de listes de diffusion ouvertes (Ling-Spam[9], PU[10] et SpamAssassin[11]) ou alors avec des messages dont les attributs ont déjà été extraits et chiffrés. Il n'y avait aucune assurance que ces messages étaient représentatifs d'une boîte aux lettres réelle.

Après la faillite de la société *Enron* en 2001, l'utilisation, pour des besoins de recherche, des messages trouvés dans les serveurs de mail de l'entreprise a été autorisée. C'est ainsi que le corpus de messages de TREC a été constitué [14]. Néanmoins, malgré l'intérêt de ce corpus, les résultats que l'on peut obtenir ne sont pas généralisables. Avec Gordon Cormack, nous avons montré [15] [16] qu'il ne pouvait pas être utilisé pour étudier les caractéristiques temporelles des filtres anti-spam. Ce corpus a également d'autres faiblesses : il a été constitué à partir de messages qui ont presque 10 ans et qui sont, pour la plupart, en texte pur (pas de partie HTML), les messages légitimes sont tous en langue anglaise, ne permettant donc pas d'étudier le filtrage dans un environnement multi-langues ou dont la langue principale n'est pas l'anglais.

[8] OpusOne Inc, - *Spam Testing Methodology* - http://www.opus1.com/www/whitepapers/spamtestmethodology.pdf - March 2007
Virus-Bulletin - *VBSpam Testing Methodology* - http://www.virusbtn.com/vbspam/methodology/index - April 2009

[9] Ling-Spam Corpus, http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz

[10] PU Corpus, http://www.aueb.gr/users/ion/data/PU123ACorpora.tar.gz

[11] SpamAssassin Corpus, http://spamassassin.apache.org/publiccorpus/

4 Métriques d'évaluation

Dans cette section, nous présentons quelques uns des indicateurs les plus pertinents pour l'évaluation de filtres anti-spam.

4.1 Tables de contingence et valeurs associées

La table de contingence (ou table de co-occurrence) est un outil souvent utilisé lorsqu'on souhaite étudier les relations entre deux variables pouvant prendre des valeurs discrètes (ou des catégories). Dans notre cas, les variables sont, dans les colonnes, le classement réel (aussi connu par *« Gold Standard »*) et dans, les lignes, le résultat du filtre. La somme de chaque colonne donne le nombre réel d'éléments dans chaque classe et celle de chaque ligne donne le nombre d'éléments vus par le classificateur dans chaque classe. Les différents rapports que l'on peut extraire de la table permettent de définir des critères d'efficacité, plus ou moins pertinents selon le type d'application.

	Vrai Ham	**Vrai Spam**
Classement Ham	9982 (VN)	334 (FN)
Classement Spam	18 (FP)	39666 (VP)

Table 1 – Exemple fictif de table de contingence présentant les résultats de classement de 10000 hams et 40000 spams

Un exemple numérique fictif de table de contingence est présenté dans la Table 1, avec les variables associées : VN (vrais négatifs : le nombre de hams vus par le filtre comme étant des hams), FN (faux négatifs : spams vus comme des hams), VP (vrais positifs : spams vus comme des spams) et FP (faux positifs : hams vus comme des spams).

Voici quelques exemples d'indicateurs que l'on peut déduire d'une table de contingence et que l'on peut trouver dans une évaluation d'un filtre anti-spam :

- ***Probabilité a priori empirique des classes*** : il ne s'agit pas d'un indicateur de l'efficacité de filtrage, mais d'un paramètre de contrôle indiquant les conditions de fonctionnement.

$$P_{spam} = (VP+FN) \;/\; (VP+FN+VN+FP)$$
$$P_{ham} = (VN+FP) \;/\; (VP+FN+VN+FP)$$

- ***Taux d'erreur par classe*** (faux positifs et faux négatifs) : il s'agit de la fraction du nombre d'objets d'une catégorie classés par erreur dans l'autre classe.

$$FPR = FP \;/\; (VN+FP)$$
$$FNR = FN \;/\; (VP+FN)$$

  Ce sont des critères pertinents et intuitifs dans les applications de classement de spams. Ils ont l'avantage de ne pas dépendre des probabilités a priori de chaque classe (répartition ham/spam).

  Pour évaluer ces taux d'erreur, certains fournisseurs utilisent la quantité totale de messages et non pas la quantité par classe : cela permet de présenter de meilleurs résultats, parfois un ordre de grandeur plus bas ;

- ***Taux de bon classement*** (vrai positifs et vrai négatifs ou sensibilité et spécificité)

$$VPR = VP \;/\; (VP+FN) = 1-FNR$$
$$VNR = VN \;/\; (VN+FP) = 1-FPR$$

- ***Précision et Rappel*** (*Precision* et *Recall*) : la *précision* indique la proportion de spams parmi les messages détectés comme étant du spam, tandis que le *rappel* est le ratio entre le nombre de spams détectés à juste titre et le nombre total de spams.

$$Precision = VP \;/\; (VP+FP)$$
$$Recall = VP \;/\; (VP+FN) = VPR$$

  Ces deux critères ont leur origine dans les applications de recherche documentaire et on les trouve parfois dans les résultats d'évaluation de filtres anti-spam, mais ils ne sont pas pertinents pour cette application, en particulier la *précision*, à cause de leur dépendance des probabilités a priori des classes ;

- ***Exactitude*** (Accuracy) : il s'agit du taux total d'erreurs, les deux classes confondues.

$$Accuracy = (VP+VN) \;/\; (VP+FN+VN+FP)$$

  Cet indicateur est souvent mentionné, mais il n'a d'intérêt que quand les classes sont plus ou moins symétriques [6], ce qui n'est pas le cas du spam où les probabilités a priori des classes, les taux d'erreurs usuels d'opération ainsi que le coût des erreurs sont souvent très différents ;

– ***Taux d'erreur pondérés*** : compte tenu de l'asymétrie des classes, plusieurs méthodes de calcul de taux d'erreur pondérés par un « coefficient de risque », associé à chaque classe ont été proposées. Actuellement le consensus est que ces indicateurs ne présentent pas d'intérêt : d'une part, malgré l'asymétrie des classes, il n'y a pas de raison pour choisir une valeur plutôt qu'une autre et d'autre part, à l'intérieur d'une même classe, le risque associé aux erreurs de classement n'est pas uniforme. Il est préférable d'afficher les valeurs pertinentes et de laisser à chacun l'interprétation.

Les indicateurs utilisant des valeurs sur les deux colonnes sont à éviter puisqu'ils dépendent de la répartition ham/spam du trafic, un paramètre qui n'est pas constant dans le temps et varie d'utilisateur à utilisateur. Nous citons la *précision* et le *rappel*, des indicateurs parfois utilisés et présentant cet inconvénient.

4.2 R.O.C. (Receiver Operating Characteristic) et 1-ROCA

Dans la section précédente nous avons vu quelques indicateurs déductibles des tables de contingence. Ces indicateurs ont l'inconvénient d'être spécifiques à un point d'opération particulier du classificateur – une valeur de seuil – et ne disent rien sur l'efficacité du filtre à d'autres points d'opération.

Certains filtres présentent le résultat de façon binaire – ham/spam (*"hard classifiers"*) tandis que d'autres sous la forme d'une valeur numérique de score (*"soft classifiers"*). Lorsque cette valeur est disponible, on peut définir des seuils tels que les messages dont le score est inférieur seront classés dans une catégorie et ceux dont le score est supérieur dans l'autre. Si l'on trace les courbes de taux d'erreurs de classement en fonction du score, on obtient une courbe similaire à celle de la Figure 3.

Une méthode simple pour obtenir le taux d'erreur en fonction du score pour tracer cette courbe est la la suivante. On classe un nombre suffisant de messages (hams et spams) et on les trie par ordre croissant de la valeur du score. Ensuite, il suffit de les parcourir et de noter pour chaque valeur de score : le nombre de spams ayant un score plus faible et le nombre de hams ayant un score plus fort[12]. Ce sont les erreurs, faux négatifs et faux positifs, que l'on obtiendrait si on avait choisi cette valeur de score comme seuil de filtrage.

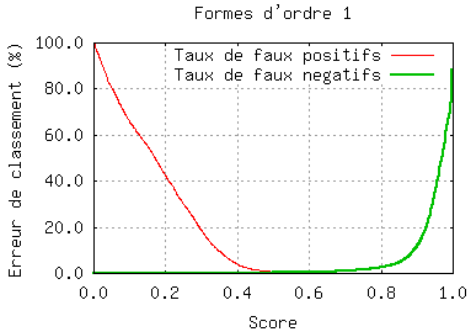

Figure 3: Taux d'erreur de classement en fonction du score

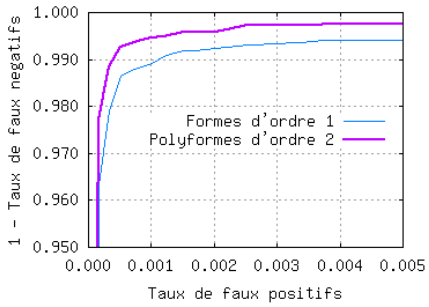

Figure 4: Courbe ROC - Taux de vrai positifs en fonction du taux de faux positifs

Le diagramme ROC (*Receiver Operating Characteristic*) [17] est un outil que l'on trouve dans des domaines tels le diagnostic médical et la radiologie, mais qui a ses origines dans les problèmes de détection radar (d'où son nom). Il s'agit de la courbe paramétrique du *taux de vrai positifs* (1 - taux de faux négatifs) en fonction du *taux de faux positifs* paramétrée par la valeur de seuil. On obtient cette courbe à partir des données utilisées pour tracer la courbe précédente.

L'exemple dans la Figure 4 présente un morceau (coin en haut à gauche) des courbes ROC de deux configurations différentes (unités de segmentation du texte) du même filtre. Pour tracer ces courbes il suffit de soumettre un ensemble de messages au classificateur et noter pour chaque valeur de score les taux de faux positifs et de faux négatifs.

Les courbes ROC ont des caractéristiques visuelles que l'on peut interpréter facilement :

– la courbe ROC est entièrement comprise dans le carré de sommets opposés (0,0) et (1,1) puisque les variables que l'on représente dans les deux axes sont des probabilités ;
– un filtre idéal (au cas où il existerait un) ne commet pas d'erreurs et donc sa courbe ROC se confond avec les côtés gauche et supérieur de ce carré ;
– le segment reliant les sommets (0,0) et (1,1) définit une zone d'incertitude où l'efficacité du filtre est identique à un choix aléatoire. Si la courbe se trouve au dessous ou à droite de ce segment, il vaut mieux faire un choix aléatoire plutôt qu'utiliser le filtre, ou alors prendre le choix inverse ;
– pour comparer deux points d'opération, il suffit de sélectionner celui que est plus vers le haut et vers la gauche ;

[12] Ce classement suppose que le score est défini de telle façon que les spams ont un score plus fort que les hams.

– pour comparer deux classificateurs, le meilleur est celui dont la courbe est le plus vers le haut et vers la gauche – dans l'exemple de la figure 4, il est préférable d'utiliser des polyformes d'ordre 2 que des formes simples.

La courbe ROC a aussi une autre propriété importante qui caractérise l'efficacité intrinsèque du classificateur. Si l'on prend, au hasard, deux objets à classer, un dans chaque catégorie, la probabilité que leurs scores soient ordonnés dans le bon ordre de classement est égale à la surface au dessous de la courbe. En général, c'est la valeur complémentaire à 1 qui est utilisée, et que l'on trouve sous les sigles (1-AUC) ou (1-ROCA). Dans l'état actuel, la valeur de ce indicateur pour les filtres anti-spam les plus performants se situe entre 0,01 et 0,05 %.

Cette courbe donne aussi un critère permettant de comparer un classificateur binaire *("hard classifier")* avec un classificateur non binaire *("soft classifier")*. Il suffit de placer le couple (VPR, FPR) dans le même graphique que celui avec lequel on souhaite comparer. Si ce point se trouve à l'intérieur de la courbe, cela veut dire qu'il est probablement moins bon que l'autre.

4.3 L.A.M. (Logistic Average Misclassification – Erreur Moyenne Logistique)

Comme nous avons vu, la courbe ROC permet d'évaluer globalement le classificateur, indépendamment du point d'opération, alors que les taux d'erreur par classe (faux positifs et faux négatifs) restent dépendants d'une valeur particulière de seuil. L'inconvénient de la ROC est le manque de relation directe entre la valeur de (1-ROCA) et les taux d'erreur par classe.

D'autre part, les taux d'erreur très faibles (inférieurs à 1 %) ont l'inconvénient d'être concentrés dans l'extrémité de l'échelle et de disparaître lorsqu'ils sont moyennés avec des valeurs d'ordre de grandeur différent. Cet inconvénient est résolu par l'utilisation du *logit* de la valeur d'erreur. La fonction *logit* et son inverse sont définies par :

$$logit(x) = \log\left(\frac{x}{1-x}\right) \quad et \quad logit^{-1}(y) = \frac{1}{1+e^{-y}}$$

Cette fonction transpose les valeurs d'erreurs, d'un intervalle fermé [0,1] vers un intervalle ouvert $(-\infty,+\infty)$ où les valeurs faibles d'erreur ne sont pas concentrées[13]. La LAM (Logistic Average Misclassification) est une estimation de l'erreur globale du filtre utilisant cette fonction pour évaluer la moyenne et est définie par :

$$LAM = logit^{-1}\left(\frac{logit(FPR)+logit(FNR)}{2}\right)$$

Il s'agit de la moyenne arithmétique que l'on calcule dans l'échelle *logit* avant de revenir à l'échelle d'origine. La valeur de la LAM est toujours intermédiaire entre le taux de faux positifs et le taux de faux négatifs et, pour des faibles valeurs, elle est proche de leur moyenne géométrique. Lors de TREC 2005, il a été remarqué [11] que la LAM varie peu avec la valeur du score, pour des valeurs faibles de (1-ROCA)[14].

Lorsqu'on n'est pas en mesure de tracer la courbe ROC parce qu'on ne peut pas mesurer les taux d'erreur pour différentes valeurs de score, on peut utiliser la propriété de faible variation de la valeur de la LAM pour estimer les taux d'erreur autour du point d'opération. Cela permet de tracer la courbe ROC et d'estimer la valeur de l'indice (1-ROCA).

4.4 Sensibilité au bruit

Pour les filtres à apprentissage, le bruit est un facteur qui peut détériorer la capacité de généralisation. Des exemples de bruit sont les retours d'informations erronées des destinataires (bruit dans le classement), des spams incluant des « mots savants » (bruit dans les attributs) ou alors, ce que l'on remarque assez souvent actuellement, la reprise d'un message généré par une *newsletter* avec remplacement des URLs et des images les plus visibles par celles du site du spammeur.

La modélisation du bruit n'est pas aussi simple, puisque les erreurs ne sont pas distribuées de façon homogène sur les différents types de messages et d'attributs. Néanmoins, la simulation d'un bruit de classement homogène permet déjà d'avoir une idée de la résistance du filtre au bruit. Pour cela, il suffit de modifier, aléatoirement, les classements des messages soumis à l'apprentissage, avec des taux entre, disons, 1 % et 10 %.

[13]Le rapport *x/(1-x)* est parfois nommé « chance » puisqu'on l'utilise, par exemple, dans les courses de chevaux : un rapport de 4 contre 1 signifie 80 % de chance de gagner et 20 % de chance de perdre.

[14]Une explication intuitive est la suivante : autour des points d'opération situés dans le coin en haut et à gauche, la courbe ROC s'approche d'une courbe hyperbolique. L'hyperbole a la propriété notable que la moyenne géométrique de ses coordonnées est constante.

4.5 Signification statistique de la mesure

L'objectif de la mesure d'efficacité d'un filtre dans une expérimentation particulière est de prédire son efficacité dans des situations similaires. La confiance que l'on peut accorder à cette mesure dépend de deux facteurs : le degré de réalisme avec lequel l'expérimentation représente les situations à prédire et les erreurs dues aux aléas de la mesure.

Ces erreurs sont données, en général, par un intervalle de confiance à 95 %, c'est-à-dire : si l'on répète la mesure N fois, il s'agit de l'intervalle qui contiendra la vraie valeur dans 95 % du temps (voir le livre de Larry Wasserman [18]). Vu que les jugements du filtre permettent d'avoir la distribution empirique des scores, une méthode adaptée à ce type d'évaluation est le « *Bootstrap* » (c'est la méthode utilisée dans TREC). Pour une description détaillée de cette méthode, voir le livre de Bradley Efron et Robert Tibshirani [19].

Outre l'évaluation de l'intervalle de confiance, il est important, autant que possible, de s'assurer que la quantité de messages à tester est suffisante pour la plage de valeurs à mesurer. Par exemple, tester un filtre sur seulement 1000 messages n'est pas suffisant si le taux de faux positifs attendu est de l'ordre de 0,1 %.

5 Évaluation hors ligne – la boîte à outils d'évaluation de TREC Spam Track

Dans les sections précédentes, nous avons décrit les contraintes à respecter lors de l'évaluation d'un filtre anti-spam ainsi que quelques paramètres intéressants à mesurer. Idéalement, si l'on veut pouvoir comparer deux filtres différents ou deux configurations différentes du même filtre, il faut pouvoir soumettre les filtres en étude aux mêmes conditions d'évaluation, si possible de façon automatisée.

Les évaluations faites dans TREC ont utilisé une boîte à outils développée par Gordon Cormack et Thomas Lynam [23] avec ces caractéristiques. Cette boîte à outils est constituée d'une partie générique de contrôle, et d'une interface spécifique à chaque filtre permettant à la partie contrôle d'envoyer des commandes au filtre et de recevoir les réponses.

Cette interface implémente quatre commandes :

- **initialize** : initialise le filtre et démarre tous les services nécessaires à son fonctionnement le rendant prêt à classifier des nouveaux messages ou à intégrer des messages au modèle d'apprentissage ;
- **classify *emailfile*** : classifie un message (fichier emailfile) et retourne le résultat : la classe (ham/spam) et le score (un nombre réel) ;
- **train *class emailfile*** : ajoute le message au modèle d'apprentissage et retourne son classement avant apprentissage ;
- **finalize** : termine le filtre, ainsi que tous les services démarrés lors de la phase d'initialisation. Après avoir envoyé cette commande au filtre, la boîte à outils évalue les critères d'efficacité à partir des résultats de filtrage.

La partie contrôle lit un fichier avec la spécification de l'évaluation, lance les actions d'évaluation dans un ordre précis, enregistre les résultats et déduit à la fin les indicateurs recherchés décrits dans ce papier, à partir des résultats du filtre en étude. La partie spécifique au filtre peut être facilement programmée dans les cas où le filtre possède une interface en ligne de commande ou des mécanismes standard de communication sous UNIX (pipes...).

Pour les filtres dont le mode de communication est le protocole SMTP, il faut intégrer un outil de soumission et de réception de messages (c'est ce qui a été fait pour le *CEAS Live Spam Challenge*).

Cette boîte à outils, distribuée sous licence GPL, peut être utilisée sans modification ou alors être personnalisée selon les besoins de chacun.

6 Conclusion

L'évaluation des filtres anti-spam est devenue un besoin aussi bien pour ceux qui développent des filtres que pour ceux qui les utilisent.

Dans cet article nous avons passé en revue les points importants à prendre en compte lors de l'évaluation d'un filtre anti-spam, ainsi que les indices d'efficacité pertinents les plus courants. Ces indices d'efficacité ne sont d'aucune valeur s'ils ne sont pas accompagnés des informations permettant de savoir comment ils ont été évalués. Certains outils de base tels la table de contingence nous permettent d'évaluer l'efficacité à des points d'opération précis, tandis que d'autres indices tels le *1-ROCA* et *LAM* sont des critères d'agrégation permettant d'avoir une idée plus globale du fonctionnement du filtre. Néanmoins, l'efficacité d'un filtre est une information à plusieurs dimensions et on ne peut pas se contenter de l'information sur un seul ou un petit nombre d'indices.

L'ensemble de ce qui a été décrit dans cet article est largement inspiré de la méthodologie d'évaluation de filtres anti-spam utilisée dans les évaluations qui ont eu lieu pendant les trois années de TREC Spam Track. L'auteur tient à remercier Gordon Cormack de l'*Université de Waterloo*, pour les nombreux échanges concernant le filtrage de spam et, en particulier, l'évaluation des filtres.

Bibliographie

[1] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. *A bayesian approach to filter junk e-mail.* In AAAI-98 Workshop in Learning for Text Categorization , 1998.

[2] W. S. Yerazunis, *The Spam-Filtering Accuracy Plateau at 99.9% Accuracy and How to Get Past It*, *MIT Spam Conference*, January 2004

[3] G. V. Cormack and T. R. Lynam,. *TREC 2005 Spam Track Overview*, In Proc. 14th Text REtrieval Conference, Gaithersburg, MD, 2005

[4] G. V. Cormack and T. R. Lynam, *Online supervised spam filter evaluation*. *ACM Transactions on Information Systems* 25, 3 (Jul. 2007), 11.

[5] J. A. Zdziarski, *Ending Spam*, No Starch Press, 2005,

[6] F. J. Provost, T. Fawcett and R. Kohavi, 1998. *The Case against Accuracy Estimation for Comparing Induction Algorithms.* In Proceedings of the Fifteenth international Conference on Machine Learning, San Francisco, CA, 1998.

[7] G. V. Cormack. *Email Spam Filtering : A Systematic Review*, volume 1. Now Publishers, 2008

[8] G. V. Cormack and A. Bratko, *Batch and Online Spam Filter Comparison,* In Proc. CEAS 2006 – Third Conference on Email and Anti-Spam, Mountain View, CA, 2006

[9] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997

[10] G. V. Cormack and T. R. Lynam, *Spam corpus creation for TREC*. in CEAS 2005 : The Second Conference on E-mail and Anti-spam, 2005

[11] G. V. Cormack and J. M. Martins da Cruz. *On the relative age of spam and ham training samples for email filtering*. In ACM SIGIR '09 : Proceedings 32nd Conference on Research and Development in Information Retrieval, New York, 2009

[12] J. M. Martins da Cruz and G. V. Cormack. *Using old spam and ham samples to train email filters*. In Proc. CEAS 2009 – Sixth Conference on Email and Anti-Spam, Mountain View, CA, 2009.

[13] A. Bratko and B. Filipic. *Spam Filtering using Character Level Markov Models: Experiments for the 2005 TREC Spam Track*, in Proc. 14th Text REtrieval Conference, Gaithersburg, MD, 2005

[14] F. Assis, *OSBF-Lua - A Text Classification Module for Lua - The Importance of the Training Method*, In Proc. 15th Text REtrieval Conference, Gaithersburg, MD, 2006

[15] G. V. Cormack, *University of Waterloo Participation in the TREC 2007 Spam Track*, In Proc. 16th Text REtrieval Conference, Gaithersburg, MD, 2007

[16] D. Sculley and G. Wachman, *Relaxed Online SVMs in the TREC Spam Filtering Track*, In 16th Text REtrieval Conference, Gaithersburg, MD, 2007

[17] T. Fawcett. *An Introduction to ROC Analysis*. Pattern Recognition Letters, 27(8) :861–874, 2006.

[18] L. Wasserman. *All of Statistics - A Concise Course in Statistical Inference*, Springer, 2004

[19] B. Efron and R. J. Tsibirani. *An Introduction to the Bootstrap*, Chapman and Hall, New York, 1994