An Overview of Content-Based Spam Filtering Techniques

Ahmed Khorsi
Department of Computer Science,
Djillali Liabes University, Bel Abbes, 22000, Algeria
E-mail: ahmed-khorsi@univ-sba.dz

Overview paper

Keywords: antispam filters, text categorization, email classification

Received: May 26, 2007

So fast, so cheap, so efficient, Internet is nowadays incontestably communication mean of choice for personal, business and academic purposes. Unfortunately, Internet has not only this beautiful face. Malicious activities enjoy as well this so fast, cheap and efficient mean. The last decade, Internet worms took the lights. In the recent years, spams are invading one of the most used services of Internet: email. This paper summarizes most of techniques used to filter spams by analyzing the email content.

Povzetek: Članek pregledno opisuje metode za filtriranje elektronske pošte.

1 Introduction

With our agreement or without, Internet is increasingly becoming a favorite support of malicious activities. After worms which are always on the foreground of information technology problems, spams appeared and are really taking day after day more intensity.

Read an E-mail is nowadays a daily habit of many people. Indeed, emails are efficient, rapid and cheap mean of communication. This makes it favorite both in professional and personal correspondences. Additionally, Reading occasionally an E-mail from unknown source and content of which is not of the user interest is not really a misfortune. However, when more than 60% or even 90% of E-mails are of such kind, and often illicit; this is what one might call a nightmare. This kind of messages is said spams.

SpamCon Inc, estimated the cost induced by productivity and resources loss, filtering software, and support caused by only one unsolicited E-mail to from 1$ up to 2$[3]; multiplied by the number of spams sent and received everyday, the one dollar becomes then millions. International Data Corp. estimates the number of spams sent everyday through the net to 7.3 billions, where only AOL users recorded 5.5 millions by March, 5, 2003 [55]. These statistics were sufficient to persuade big users of the E-mail service to forecast a supplementary budget to fight spams. UUNet, one of the most important ISPs has a group of six persons with a budget of 1 million dollars, just to fight spams [56]. Netcom estimated that 10% of the end-user invoice is dedicated to filter spams [54]. A study of International Data Corporation (IDC) ranked spams in the second position of the ISPs' problems. One question arises then; *Why does someone enjoy sending so many E-mails, and how does he get so many addresses*? Although motivations are sometimes different, spams are generally of publicity-like contents. To broadcast a commercial through a TV channel costs hundreds times more than sending millions of spams. To get so many E-mail addresses is not at all difficult since many are available in the Internet itself. Some spammers, uses addresses found in newsgroups publicly accessible. Some others use webbots commonly called spambots, software which browses automatically the web seeking E-mail addresses. Generally, Spambots use keywords matching techniques to extract the email addresses. One evident way is to check for the character ''. Some others use software to generate random addresses then record all addresses from which they do not receive a reply of a delivery failure. More advanced techniques are summarized in [57].

Actually, fighting spams takes various forms. Juridical one came early in US by adopting an anti-spam law [30]. International cooperation is growing as well [47]. Simpler method might consist in some good practi-

	Computer viruses	**Spams**
1.	their authors are human programmer who tempt to workaround anti-viruses.	Senders are humans who tempt to avoid characteristics which denounce their spams.
2.	tempt to infect as many as possible systems.	spammer tempt to send their messages to as many as possible Internet users.
3.	may cause lot of damage in the infected system.	Consumes a large part of the Internet bandwidth, and causes a considerable productivity loss.
4.	it introduce itself discreetly in systems.	they are generally unsolicited

Table 1: Similarities between viruses and spasm.

ces. For instance, do not publish textual E-mail addresses, and use images instead, do not reply to suspicious E-mails. Some other techniques use features of headers such as E-mail origin [58][18][61]. These methods are widely supported by the most used Email servers without need of filtering software. However, all these methods are not the purpose of our discussion.

An antispam filter is similar to an anti-virus which scans files to check for virus signatures. Indeed, there are many similarities between computer viruses and spams. Table 1 enumerates some of them. In the following

Actually, many of the filtering techniques are based on text categorization methods [28] [8]. Thus filtering spams turns on a classification problem. Roughly, we can distinguish between two methods of machine classification. The first one is done on some rules defined manually. The typical example is the rule based expert systems. This kind of classification can be used when all classes are static, and their components are easily separated according to some features. The second one is done using machine learning techniques [23]. It is more convenient when the characteristics of discrimination are not well defined. These techniques attempt to generate on a set of samples, quasi or semi automatically a classifier with an acceptable error rate.

2.1 Generalities

Getting an E-mail *m*, we have to define a decision function *f* which assigns to *m* its class, *S* for Spams or *L* for legitimate. Let G_M be the set of messages. *f* is then :

$$f : G_M \rightarrow \{S,L\}$$

In such techniques, we first check some characteristics on which we may classify the message into the class *S* or *L*. We will refer to such characteristics by a vector *x*.

Let *P(x/c)* be the probability that the class *c* generates a message which its characteristic vector is *x*. If we suppose that a legitimate message never contains the text *t* ="*Buy now*" and that *x = (m = utv)* with *u*, *v* are two strings, the probability *P(x/L) = 0*. Then, the problem is to compute the probability that a message which has a characteristic vector *x* belongs to the class *c* say *P(c/x)*. We obtain then by observing the rule of Bayes:

$$P(c/x) = \frac{P(x/c)P(c)}{P(x)} = \frac{P(x/c)P(c)}{P(x/S)P(S) + P(x/L)P(L)}$$

Where *P(x)* indicates the a-priori probability of occurrence of a message which characteristic vector is *x*, and *P(c)* indicates the probability that a random Email belongs to the class *c*. Knowing the probability *P(c)* and the probability *P(x/c)* suffices to deduce *P(c/x)*. We have then the following rule of classification:

If *P(S/x) > P(L/x)* (the a-posteriori probability that the E-mail which has the characteristic vector *x* belongs to the class *S* is greater than that the same E-mail belongs to the class *L*) then classify *m* as being unsolicited.

sections we will briefly present some content-based filtering techniques. The main idea behind such techniques is to classify an email into unsolicited or legitimate by checking some features in its content. It is not our aim to make an extensive comparison of these different methods when many papers propose comparisons between two or more of these techniques[34][20][2][60]

2 Bayesian classifier

This rule is called the rule of the maximum aposteriori probability(MAP). It can be written as follow: If

$$\frac{P(x/S)}{P(x/L)} > \frac{P(L)}{P(S)}$$

classify the message as being unsolicited and legitimate otherwise.

We often note the resemblance fraction:

$$\Lambda(x) \quad \frac{P(x/S)}{P(x/L)}$$

The MAP rule is written then:

$$\Lambda(x) \underset{L}{\overset{S}{\gtrless}} \frac{P(L)}{P(S)}$$

Let us note $L(c_1, c_2)$ the function which determines the cost of a bad classification of an occurrence of the class c_1 as being of the class c_2. It is logic to say whereas $L(L,L) = L(S, S) = 0$. We can then define a function of risk:

$R(c/x) = L(S, c)P(S/x) + L(L, c)P(L/x)$

Obviousness would be to classify the message by minimizing the function of risk. From which the rule:

If R(S/x) < R(L/x) classify m unsolicited and legitimate otherwise.

This last rule is called bayesian rule of classification or Bayes classifier[44].

We write the classification rule in term of resemblance fraction as follow:

$$\Lambda(x) \underset{L}{\overset{S}{\gtrless}} \lambda\frac{P(L)}{P(S)}$$

Where

$$\lambda = \frac{\mathcal{L}(L,S)}{\mathcal{L}(S,L)}$$

Intuitively, this parameter indicates the risk taken when we classify a legitimate E-mail as being unsolicited. Clearly, more λ is great; more the false positive error is small.

2.2 Application

In this subsection, we highlight the practical application of the theoretical principle of the bayesian classifier.

As already mentioned, to be able to determine the classification parameter, one must determine the probabilities *P(x/c)* and *P(c)* for any message *m*. Obviously, that cannot be made in exact manner. However, we can approximate these probabilities on the basis of a training sample. For example, the probability *P(S)* would be roughly given by calculating the ratio of the spams on the number of all messages in the training sample.

For simplicity, we consider that the characteristic vector is a binary one, where the presence of a catchword w in the message *m* is represented by one 1. That is to say then:

$$P(x_w = 1/S) \approx \frac{\text{Number of spams in which} w \text{ is checked}}{\text{number of all spams}}$$

Algorithm 1 summarizes the training where Algorithm 2 the classification steps.

Algorithm 1 Training algorithm of Bayes's classifier

1: **for all** $c \in \{S, L\}$ **do**
2: Estimate $P(c)$
3: **for all** $x_w \in \{0, 1\}$ **do**
4: Estimate $P(x_w/c)$
5: **end for**
6: **end for**
7: **for all** $x_w \in \{0, 1\}$ **do**
8: Calculate $P(c/x_w)$ using Bayes's rule
9: **end for**
10: **for all** $x_w \in \{0, 1\}$ **do**
11: Estimate $\Lambda(x_w)$
12: **end for**
13: Calculate $\lambda \frac{P(L)}{P(S)}$

Algorithm 2 Classification based on Bayes's classifier

1: Calculate the vector x_w of the input message m
2: **if** $\Lambda(x_w) > \lambda \frac{P(L)}{P(S)}$ **then**
3: m is unsolicited
4: **else**
5: m is a legitimate email
6: **end if**

In general, we represent the presence of a word w_i in the message *m* by the value 1 in the characteristic vector $x = (x_1, x_2, \ldots, x_n)$. However, the algorithm 1 will have to compute 2^n values of x which is unpractical. To avoid this, the assumption is introduced that the presence of two words is independent one of the other, which allows us to write:

$$P(x/c) = \prod_{i=1}^{n} P(x_i/c) \qquad \Lambda(x) = \prod_{i=1}^{n} \Lambda_i(x_i)$$

In [32] T.A Meyer and B Whateley report that using foure corpora gathered from SpamBayes users and SpamAssassin public corpus they obtained the following results. The bayesian classifier constitutes one of the most used techniques in antispam filters such as ' *spamassassin*' [53] and SpamBayes [32] or [49]. Although the assumption of mutual independence between word's occurrences is false, the recorded results remain very good for the traditional text messages. It often serves as a baseline to compare performances of other methods [60].

3 *k* nearest neighbors

The principle of this technique is rather simple. Let us suppose that similarities among messages are measurable using a measure of distance among the characteristic vectors. To decide whether a message is legitimate or not, we look at the class of the messages that are closest to it. Generally, this technique does not use a separate phase of training and the comparison between the vectors is a real time process. This has a time complexity of *O(nl)* where n is the size of the characteristic vector and l the sample size. This can be circumvented by using a traditional indexing methods [13][19][35]. To adjust the risk of false classification, t/k rule is introduced. What can be read:

> If at least *t* messages in *k* neighbors of the message *m* are unsolicited, then *m* is unsolicited email, otherwise, it is legitimate.

We should note that the use of an indexing method in order to reduce the time of comparisons induces an update of the sample with a complexity *O(m)*, where *m* is the sample size. An alternative of this technique is known as memorybased approach [2][46].

TiMBL [11] is a software package developed by ILK Research Group that implements a collection of machine learning algorithms. Results of the implementation of this technique in spam filtering reported in [2] seems to be comparable to those of bayesian classifiers.

4 Technique of Support Vector Machine (SVM)

Support vector machine [9][10][7] is one of the most recent techniques used in text classification. In machine learning the training sample is a set of vectors of n attributes. We can then assume that we are in a hyper-space of *n* dimensions, and that the training sample is a set of points in the hyper-space. Let us consider the simple case of just two classes (as it is the case of spam problem). The classification using Support vector machine look for the hyper plane able to separate the points of the first class from those of the second one such that the distance between the hyper plane and points of each class is maximum see Figure 1.

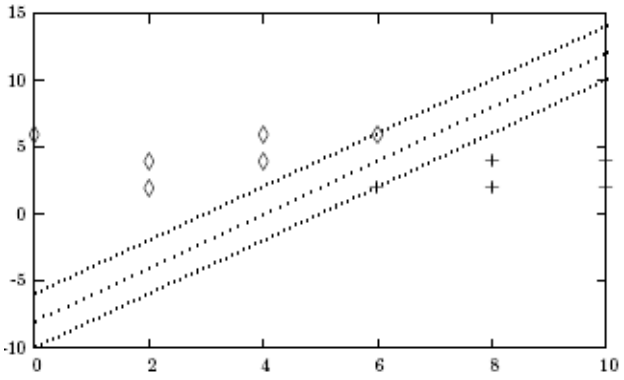

Figure 1: hyper-plane that separate two classes.

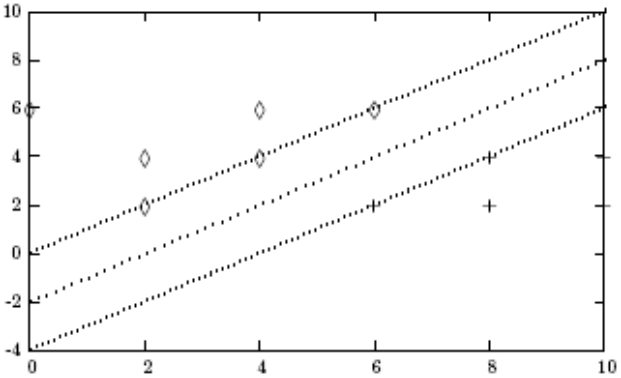

Figure 2: hyper-plane that separate two classes and is far from each class.

One question may be how we can find the hyper plane when the classes are not linearly separable (eg. XOR function). In this case the hyper space is extended to more dimensions. This insures the existence of hyper plane that separates the two classes. One interesting feature is that to find the appropriate plane, SVM method explore just the nearest points. One of the most efficient SVM algorithms was proposed in [39]. An implementation of the SVMmethod in spam filtering is proposed in [12] where Dricker et all also provide also a comparison with other methods.

5 Technique of maximum entropy

Maximum entropy is a classical model often used in natural language processing [41]. The principle is to find the appropriate probability distribution *p(a, b)* that maximizes the entropy:

$$H(p) = -\sum_{x \in \mathcal{A} \times \mathcal{B}} p(x) \log p(x)$$

where $\mathcal{A}$ denotes the set of possible classes, and $\mathcal{B}$ the set of possible values of vectors of features. This maximization should keep *p* consistent with evidence (i.e., should meet all known values in the training set). *p* becomes is then:

$$p(a,b) = \frac{1}{Z(b)} \prod_{j=1}^{k} \alpha_j^{f_j(a,b)}$$

where *k* is the size of the vector of features and

$$Z(b) = \sum_{a} \prod_{j=1}^{k} \alpha_j^{f_j(a;b)}$$

is a normalization factor that ensures

$$\sum_a p(a,b) = 1.$$

α_j can be computed using the Generalized Iterative Scaling [15]. *f* is defined as follows:

$$f_{cp,a'}(a,b) = \begin{cases} 1 & \text{if } a = a' \text{ and } cp(b) = true \\ 0 & \text{otherwise} \end{cases}$$

where cp maps a pair *(a, b)* to *{true, false}* Results reported in [59] show an error rate better than that of bayesian classifier when the training sample grows.

6 Technique of neural networks

Neural network is a well known model [51][31][42][14][17][17][50] which has been designed by McCuloch on the basis of work carried out on the human neurons. The neural networks are quite famous to be well adapted for problems of classification. Without being spread out over the model, we will retain in what follows the characteristics which contribute to the design of an antispam filter.

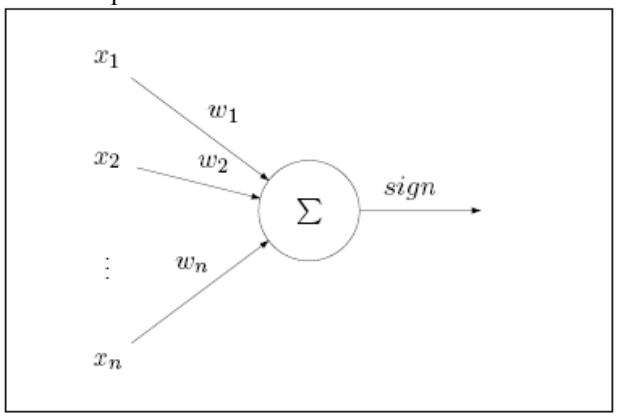

Figure 3: The perceptron.

6.1 Perceptron

The idea is to define a linear function *f(x) = wx + b* where:

$$f(x) \begin{cases} > 0 & \text{If } x \text{ is of the first class} \\ < 0 & \text{else} \end{cases}$$

where *w* is a vector of weights and *b* a bias vector. We can simplify the function to obtain a decision function from it *d(x) = sign(wx + b)*[43]. Figure 3 shows a graphical representation of the perceptron. The training of the perceptron is performed using an iterative method, where the weight and bias vectors are initialized then adjusted each iteration in such manner to ensure the classification of a new occurrence of the training sample. For instance let *x* be a vector that the perceptron fails to classify, and w_i, b_i the vector of weight and bias which corresponds to the i^{th} iteration. We have $sign(w_i x+bn) \neq c$

where c is the sign corresponding to the real class of the message that has the characteristic vector x. The new vectors w_i+1 and b_i+1 are calculated as follow:

$$w_{i+1} = w_i + cx \qquad b_{i+1} = b_i + c$$

The training continues until the perceptron manages to classify correctly all the messages of the training sample. In this case, we say that the perceptron converges. It is well-known that the perceptron does not converge in the case of non-linear classification problem [21][16].

In the case of spams filtering and if one makes a point of applying the technique of the perceptron, it is enough to choose a characteristic vector larger than that of the training sample to ensure the convergence. However such practice will heavily weigh down the computation.

The algorithm 3 summarizes the training of the perceptron.

Algorithm 3 Training algorithm of the perceptron

1: Initialize w and b.
2: **while** $\exists x \in$ training sample such that $sign(wx + b) \neq c$ **do**
3: $w \leftarrow w + cx$ and $b \leftarrow b + c$
4: **end while**

6.2 The multi-layer networks

As its name indicates, the multi-layer neural net is a network of connected perceptrons which form a network with successive layers. The outputs of each perceptron are inputs of perceptrons of the following layer. The inputs of the neurons of the first layer are the components of the characteristic vector, while the outputs of the last layer are the results of the classification. The layers between the first and the last are called *hidden layers*. The function of each neuron is somewhat different from the simple perceptron, although the training is also made in an iterative way as the simple perceptron. The output function is:

$$o = \phi(\sum_{i=1}^{k} w_i x_i + b)$$

where ϕ is a nonlinear function such as

$$\frac{1}{1+e^{ax}}$$

Or tanh(x). Figure 4 shows a graphical representation of a multi-layer neural network. The training of the neural network means the readjustment of the weights and bias in such manner to minimize the sum of the errors of the output, that is to say:

$$E(f) = \sum_{i=1}^{n} |f(x_i) - c_i|^2$$

The tuning of these parameters is described in details in [21][16]. In [38] Levent ¨Osg¨ur and all reported a 90% accuracy in a filter based on coupling neural network technique and bayesian classifier. [33] is a Semantec white paper on how and why neural network should be implemented in an antispam system.

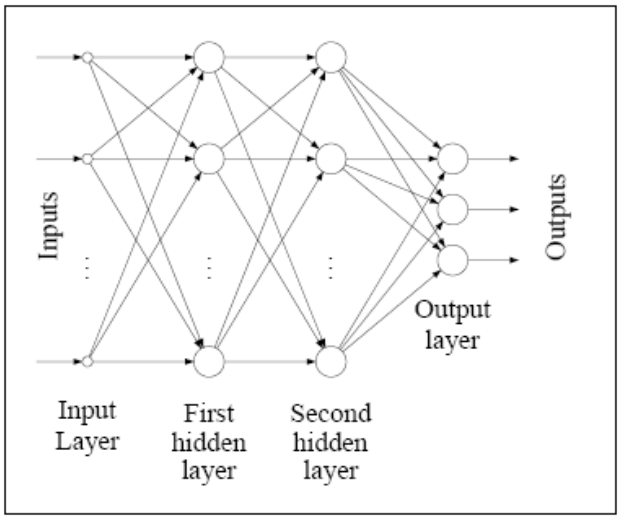

Figure 4: Multi-layer neural network.

7 Technique of search engines

When it acts on text e-mails, classification techniques of text seem to be efficient. However, spammers do not cease to invent tricks to circumvent filters. One of these tricks is to include in the body of the message only the hyperlink to a Web page which contains the advertising text. The problem become then a web content classification. A proposed technique to overcome this kind of spams is to use the public search engines which offer a mean to classify the websites [22]. The principle of this technique is to analyze automatically the contents of the pages referred by the links sent in the messages likely to be spams. The analysis starts by using the public search engines such as Yahoo and Google. A comparison then with the user interest can judge the convenience of the message with the requests of the user. If the search engines do not label the referred pages, a later step consists to analyze contents of the pages by traditional Bayes's classifier. Initial classification by the search engines is also used to enrich the sample of the bayesian classifier. This makes the model more dynamic. The main drawback of this approach is that to judge whether a mail is legitimate or not its important use of the band-width.

In [22], the author reports a false positive rate of 0.0032% on a sample of 1191 legitimate emails and 493 spams.

Type	Operation	Description
Arithmetic	+,-,/,*,$\sqrt{}$	
Relational	=,$\neq$,$\geqslant$,¡	
Boolean	AND,OR,NOT	
Non-linear	max,min,ABS	
Of words	Freq(x)	return the frenquency of the word$x$ in the message body
	Exists(x)	Return 1 if the word x occurs in the message and 0 otherwise

Table 2: Operation represented in the tree.

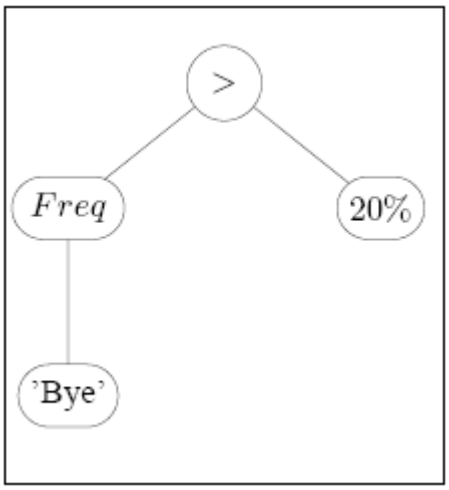

Figure 5: An example of a tree of a bayesian classifier.

8 Technique of genetic programming

In the design of a bayesian filter, the characteristic vector may include the frequencies of some words [45] generally selected by human experts. In fact, this construction is sometimes decisive in the performances of the filter. In [20], Hooman proposes a method to build automatically the bayesian filter. This method is based on the genetic programming. Thus, the frequency of a word ' *buy* ' for example ' *60*' % in an E-mail can argument the classification of the message as unsolicited. As genetic programming suggested by Koza [25][26][24][27][1], the filter is represented by a syntactic tree where nodes are :

- numbers that represent the frequencies
- operations on numbers
- words
- operations on words

see Table 2.

A syntactic tree of a filter should be built according to a precise syntax. Syntactic rules then can be used to check the correctness of the tree by checking whether we are able to reduce the tree to some number. Figure 5 gives an example of a tree of a filter. Artificial genetic approaches use functions that evaluate the fitness of a population to some criteria. This is also the case of the approach being explained.

The fitness function is then defined as follows:

$$Fitness = \frac{1}{|m_S|}\sum_{i=1}^{|m|} S(m_i)(v_i - a_i)^2 + \frac{1}{|m_L|}\sum_{i=1}^{|m|} L(m_i)(v_i - a_i)^2 \quad (1)$$

Where

a_i □ *{0, 100}* Is the correct classification of the message m_i

$0 \leq vi \leq 100$ the classification of mi returned by the filter.

m_S The number of spams.

m_L The number of legitimate E-mails.

$S(m_i)$ returns 1 if m_i is a unsolicited and 0 otherwise $L(m_i)$ returns 1 if mi is legitimate and 0 otherwise

Figure 6 gives an example of cross-over. According to the author of this approach, the experimental results reported in [20] shows an effectiveness close to those of a bayesian classifier manually built

9 Technique of Artificial Immune System

The almost obvious similarities between spams and computer viruses let think that the traditional and new techniques of anti-viruses can be applied to fight spams. One of these techniques is computer immune systems [36][52][48]. In [37] Terri Oda and Tony White suggest to design an anti-spams filter based on the generation of artificial lymphocytes using gene database. Genes are regular expressions which represent mini-languages likely to contain keywords that are usually checked in spams. The use of the regular expressions aims according to the author at increasing the accuracy as well as the general information hold in the detecting lymphocytes. The generation of lymphocytes is based on a training sample. The lifespan of these lymphocytes can be tuned

in order to ensure the system dynamicity. This technique represents only an attempt to assist the classical techniques already proved. Membranar proteins of biological cells allow a deterministic way to check whether a cell is self or not. It remains too difficult to find efficient discrimination between viruses and legitimate objects of computer systems as well as between spams and legitimate Emails

10 Conclusion

It is now well known that no technique can be claimed alone to be the ideal solution with 0% false positive and 0% false negative. Currently used antispam systems couples several machine learning techniques for content classification. Spamassassin uses the genetic programming to generate its bayesian classifier for each release. Text classification techniques, such as bayesian classifiers and neural networks offer a good theoretical and practical background to fight the problem of spams. However, two disadvantages are opposed to such relatively simple approaches. First, the definition of unsolicited E-mails varies from one to another. A generalized classification can penalize some users interested by some products advertised electronically. The second disadvantage is that a mail can be other thing than simple text. Take the example of the multimedia messages (images, voice-email, and movies). If methods of text classification are allowed to wander the text of each message, wandering tens of thousands of images or movies to classify them is surely not a practical solution. A solution of the first problem is to base classification on user profiles rather than impose characteristic vectors issued from perceptions other than those of the users. More general solution would be an hierarchical filter. In each node of the hierarchical tree the filter should block all E-mails which seem to be unsolicited from the users of all its sub-trees. Regarding the second disadvantage, the methods of classifications of multimedia documents exist [40][6][29] but their time and space complexities remain far from the requirements of a real time computation. Recently, new approches which count links of spam have been also investigating [4][5] Fighting spams is series of chess parts between industrial

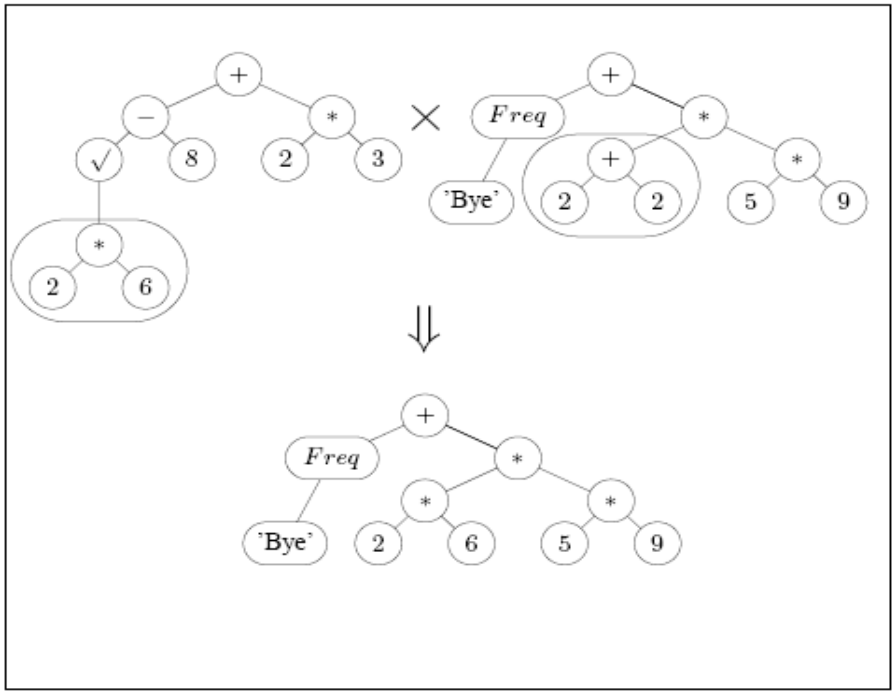

Figure 6: Example of a cross between two trees

and researchers in one side and spammers in the other side. Until the day the latter will decide to do not play any more, researchers will abstain from shouting victory, just like industrials and researchers of the anti-viruses. Spams may be a misfortune for simple users, but it seems to be a new big market for information technology industrials.

Acknowledgement

My thanks go to Dr. Bendahmane, Dr. Djelloul Ziadi, and all persons who contributed to this work.

References

[1] M. Ahluwalia and L. Bull. A genetic programming-based classifier system. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 11–18, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

[2] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memorybased approach. In H. Zaragoza, P. Gallinari, , and M. Rajman, editors, *Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, pages 1–13, 2000.

[3] S. Atskins. Size and cost of the problem. In *Priceedings of the Fifty-sixth Internet Engineering Task Force(IETF) Meeting*, (San Francisco, CA), March 16-21 2003. SpamCon Foundation.

[4] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. Technical report, DELIS – Dynamically Evolving, Large-Scale Information Systems, 2006.

[5] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. In *Workshop: The Future of Web Search*, Barcelona, May 2006.Universitat Pompeu Fabra.

[6] J. S. De Bonet. Novel statistical multiresolution techniques for image synthesis, discrimination, and recognition. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1997.

[7] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[8] W. W. Cohen. Learning to classify English text with ILP methods. In Luc De Raedt, editor, *Advances in inductive logic programming*, pages 124–143. IOS Press, Amsterdam, NL, 1995.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[10] N. Cristiatnini and J. Shawe-Taylor. *An introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2003. `http://www.support-vector.net`.

[11] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. Timbl: Tilburg memory based learner, version 1.0, reference guide. ILK Technical Report 98-03, Tilburg, 1998.

[12] H. Drucker, V. Vapnik, and D. Wu. Support vector machines for spam categorization. I*EEE Transactions on Neural Networks*, 10(5):1048–1054, 1999. Available: `http://www.monmouth.edu/ druker/SVM spam article compete.PDF`.

[13] P. Ferragina and R. Grossi. Improved dynamic text indexing. *J. Algorithms*, 31(2):291–319, 1999.

[14] M. Glick and D. Rumelhart. *Neuroscience and Connectionist Theory*. The Developpement in Connictionist Theory. Erlbaum Assiciates, Hilldsdale,

[15] NJ, 1989.J. Goodman. Sequential conditional generalized iterative scaling. In *ACL '02: Proceedings of the40th Annual Meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[16] S. Haykin. *Neural Networks: A Fomprehensive Foundation*. Printice Hall, 1998.

[17] J. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.

[18] G. Hulten, J. Goodman, and R. Rounthwaite. Filtering spam e-mail on a global scale. Technical report, Microsoft Corp, 2004.

[19] J. K¨arkk¨ainen and E. Ukkonen. Lempel-ziv parsing and sublinear-size index structures for string matching. In *Proc WSP'96*, pages 141– 155. Carleton University Press, 1996.

[20] H. Katirai. Filtering junk e-mail: A performance comparison between genetic programming and naive bayes.

[21] V. Kecman. *Learning and Soft Computing*. The MIT Press, 2001.

[22] O. Kolesnikov, W. Lee, and R. Lipton. Filtering spam using search engines. Technical Report GIT-CC-04-15, Georgia Tech, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, 2004-2005.

[23] A. Konar. *Artificial Intelligence and Soft Computing : Behavioral and Congnitive Modeling of the Human Brain*, chapter 8. CRC Press, washington, 2000.

[24] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, MA, 1992.

[25] J. R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In N. S. Sridharan, editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89*, volume 1, pages 768–774. Morgan Kaufmann, 20-25 aug 1989.

[26] J. R. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Dept. of Computer Science, Stanford University, June 1990.

[27] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA, 1994.

[28] D. Lewis. (spam vs.) forty years of machine learning for text classification. In *Proceedings of the Spam Conference*, 2003.

[29] L. Lu, H. Jiang, and H. Zhang. A robust audio classification and segmentation method. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 203–211, New York, NY, USA, 2001. ACM Press.

[30] D. Platt Majoras, O. Swindle, Commissioner, T. B. Leary, P. O. Harbour, Commissioner, and J. Leibowitz. A can-spam informant reward system : A report to congress. Technical report, Federal Trade Commission, US, 2004.

[31] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity.

Bulletin of Mathematical Biophysics, 5:115– 133, 1943.

[32] T. A. Meyer and B. Whateley. Spambayes: Effective open-source, bayesian based, email classification system. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004. Available: http://www.ceas.cc/papers-2004/136.pdf.

[33] C. Miller. Neural network-based anti-spam heuristics, white paper.

[34] K. Mock. An experimental framework for email categorization and management. In *24th Annual ACM International Conference on Research and Development in Information Retrieval*, New Orleans, LA, September 2001.

[35] G. Navarro and R. Baeza-Yates. A practical q-gram index for text retrieval allowing errors. *CLEI Electonic Journal*, 1(2), 1998.

[36] G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, editors. *Artificial Immune Systems, Third International Conference, ICARIS 2004, Catania, Sicily, Italy, September 13-16, 2004*, volume 3239 of *Lecture Notes in Computer Science*. Springer, 2004.

[37] T. Oda and T. White. Developing an immunity to spam. In *GECCO*, pages 231–242, 2003.

[38] L. ¨Osg¨ur, T. G¨ung¨or, and F. G¨urgen. Adaptive anti-spam filtering for agglutinative languages: a special case for turkish. *Pattern Recogn. Lett.*, 25(16):1819–1831, 2004.

[39] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[40] *29th Applied Image Pattern Recognition Workshop (AIPR 2000),16-18 October 2000, Washington, DC, USA, Proceedings*. IEEE Computer Society, 2000.

[41] A. Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, University of Pennsylvania, 1997.

[42] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, Washington, 1958.

[43] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. In J. W. Shavlik and T. G. Dietterich, editors, *Readings in Machine Learning*, pages 138–149. Kaufmann, SanMateo, CA, 1990.

[44] M. Sahami. Learning limited dependence Bayesian classifiers. In *Second International Conference on Knowledge Discovery in Databases*, 1996.

[45] M. Sahami. A bayesian approach to filtering junk e-mail. In *Proceedings of AAAI-98 workshop on Learning for Text Categorization*, Madison, Wisconsin, USA, 1998.

[46] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. A memory-based approach to antispam filtering for mailing lists. *Information Retrieval*, 6:49–73, 2003.

[47] C. Sarrocco. Spam in the information society: Building frameworks for international cooperation. Technical report, OECD Task force on spam, 2004.

[48] A. Secker, A Freitas, and J. Timmis. Aisec: An artificial immune system for e-mail classification. In R. Sarker, R. Reynolds, H. Abbass, T. Kay-Chen, R. McKay, D Essam, and T. Gedeon, editors, *Proceedings of the Congress on Evolutionary Computation*, pages 131–139, Canberra. Australia, December 2003. IEEE.

[49] S. Shakeri and P. Rosso. The bsp spam filter. In *Proc. Confer. Information Technologies Int. Symposium*, Tutan, Morocco, June 2005. IEEE.

[50] J. Shavlik, R. Mooney, and G. Towell. Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning*, 6:111–143, 1991.

[51] H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

[52] A. Somayaji, S. Hofmeyr, and S. Forrest. Principles of a computer immune system. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms*, pages 75–82, New York, NY, USA, 1997. ACM Press.

[53] Spamassassin. Spamassassin website. http://spamassassin.org.

[54] New York Times, March 19, 1998.

[55] J.Weaver. Aol escalates spam warfare, March 5, 2003.

[56] Internet Week, May 4, 1998.

[57] Gregory L. Wittel and S. Felix Wu. On attacking statistical spam filters. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[58] H. K. Yoke. Curbing spam via technical measures: an overview. Technical report, ITUWSIS Thematic Meeting on Countering Spam, 2004.

[59] L. Zhang and T. Yao. Filtering junk mail with a maximum entropy model. In *Proceeding of 20th International Conference on Computer Processing of Oriental Languages (ICCPOL03)*, pages 446–453, 2003.

[60] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACMTransactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.

[61] F. Zhou, L. Zhuang, B. Y. Zhao, L. Huang, A. D. Joseph, and J. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Proc. of ACM/IFIP/USENIX Intl. Middleware Conf*, 2003.