UNIVERSITY
OF TRENTO - Italy

**Information Engineering
and Computer Science Department**

A SURVEY OF LEARNING-BASED TECHNIQUES OF EMAIL SPAM FILTERING

Enrico Blanzieri and Anton Bryl

January 2008 (Updated version)

Technical Report # DIT-06-056

A Survey of Learning-Based Techniques of Email Spam Filtering

Enrico Blanzieri,
University of Trento, Italy,
and
Anton Bryl
University of Trento, Italy,
Create-Net, Italy
anton.bryl@dit.unitn.it

January 11, 2008

Abstract

Email spam is one of the major problems of the today's Internet, bringing financial damage to companies and annoying individual users. Among the approaches developed to stop spam, filtering is an important and popular one. In this paper we give an overview of the state of the art of machine learning applications for spam filtering, and of the ways of evaluation and comparison of different filtering methods. We also provide a brief description of other branches of anti-spam protection and discuss the use of various approaches in commercial and non-commercial anti-spam software solutions.

1 Introduction

The problem of undesired electronic messages is nowadays a serious issue, as spam constitutes up to 75–80% of total amount of email messages [63]. Spam causes several problems, some of them resulting in direct financial losses. More precisely, spam causes misuse of traffic, storage space and computational power [86]; spam makes users look through and sort out additional email, not only wasting their time and causing loss of work productivity, but also irritating them and, as many claim, violating their privacy rights [86]; finally, spam causes legal problems by advertising pornography, pyramid schemes, etc. [68]. The total worldwide financial losses caused by spam in 2005 were estimated by Ferris Research Analyzer Information Service at $50 billion [31].

Lately, Goodman et al. [39] presented an overview of the field of anti-spam protection, giving a brief history of spam and anti-spam and describing major directions of development. They are quite optimistic in their conclusions, indicating learning-based spam recognition, together with anti-spoofing technologies and economic approaches, as one of the measures which together will probably lead to the final victory over email spammers in the near future. Presently, according to the study by Siponen and Stucke [86] about the use of different kinds of anti-spam tools and techniques in companies, filtering is the most popular way of protection from spam. This shows that spam filtering is, and is likely to remain, an important practical application of machine learning.

In this paper we give a structured overview of the existing learning-based approaches to spam filtering. One section describes the spam phenomenon, including a brief overview of non-filtering techniques, which we think is necessary for understanding the context in which a spam filter works. Our survey gives a systematic guide to the present state of the literature, considering a wide scope of papers, and being thus complementary to the work of Goodman et al. [39],

who present a concise account of the history of anti-spam protection and the directions of future development. An overview of email classification, including spam filtering, was previously given by Wang and Cloete [93]. Compared to their work, we overview a much wider variety of filtering techniques and pay more attention to evaluation and comparison of different approaches in the literature.

The survey does not intend to cover neighboring topics, being devoted to protection from email spam. In particular, we do not address the issue of viruses delivered by spam, because we believe that this two problems, namely spam and viruses, are always distinguishable enough to be discussed separately: a virus can be recognized as such without reference to the way of delivery of it, and a spam message can be recognized as such both with and without malicious content. Also, we focus on the email spam, not on spam in general. Though the spam delivered through instant messengers, blog comments or systems of voice transmission pursues similar goals, the technical differences are significant enough to make the problem of spam in general too complex for one overview (see, for example, the paper by Park et al. [72] for discussion of differences between email and voice spam).

The paper is organized as follows: Section 2 is an introduction to the phenomenon of spam, including a brief overview of anti-spam efforts not based on filtering; Section 3 is dedicated to the methods of machine learning used for spam filtering; Section 4 is a brief glance on the existing commercial and non-commercial software solution; Section 5 overviews evaluation and comparison methods; finally, Section 6 is a conclusion.

2 The Spam Phenomenon

This section provides an introduction to the phenomenon of spam, including the definition and general characteristics of spam, as well as a brief overview of non-filtering methods of anti-spam protection, namely anti-spam legislation and changes in the process of email transmission. Not being directly related to spam filtering, this methods either influence the ways in which spam can be formed and transmitted, or provide new architectures in which a filter can be used. Therefore, a brief introduction to this methods is needed before passing to filtering itself.

2.1 Definition and General Characteristics of Spam

There exist various definitions of what spam (also called junk mail) is and how it differs from legitimate mail (also called non-spam, genuine mail or ham). The shortest among the popular definitions characterizes spam as "unsolicited bulk email" [3, 90]. Sometimes the word *commercial* is added, but this extension is argued. The TREC Spam Track relies on a similar definition: spam is "unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the user" [19]. Another widely accepted definition states that "Internet spam is one or more unsolicited messages, sent or posted as part of a larger collection of messages, all having substantially identical content" [88]. Direct Marketing Association proposed to use the word "spam" only for messages with certain kinds of content, such as pornography, but this idea met no enthusiasm, being considered an attempt to legalize other kinds of spam [89]. As we can see, the common point is that spam is *unsolicited*, according to a widely cited formula "spam is about consent, not content" [90]. It is necessary to mention that the notion of being unsolicited is hard to capture. In fact, despite the wide agreement on this type of definitions the filters have to rely on content and ways of delivery of messages to recognize spam from legitimate mail. Among the latest work it is interesting to mention Zinman and Donath [106], who still prefer to rely on content and a user's personal judgement to define spam.

There is a growing scientific literature addressing the characteristics of the spam phenomenon. In general, spam is used to advertise different kinds of goods and services, and the percentage of advertisements dedicated to a particular kind of goods or services changes over time [46]. Quite often spam serves the needs of online frauds. A special case of spamming

activity is *phishing*, namely hunting for sensitive information (passwords, credit card numbers, etc.) by imitating official requests from a trusted authorities, such as banks, server administration or service providers [24]. Another type of malicious spam content are viruses [61]. Sometimes a massive spam attack can be used also to upset the work of a mail server [69]. To sum up, the sender of a spam message pursues one of the following tasks: to advertise some goods, services, or ideas, to cheat users out of their private information, to deliver malicious software, or to cause a temporary crash of a mail server. From the point of view of content spam is subdivided not just into various topics but also into several genres, which result from simulating different kinds of legitimate mail, such as memos, letters, and order confirmations [21]. Characteristics of spam traffic are different from those of legitimate mail traffic, in particular legitimate mail is concentrated on diurnal periods, while spam arrival rate is stable over time [35]. Spammers usually mask their identity in different ways when sending spam, but they often do not when they are harvesting email addresses on websites, so recognition of harvesting activities can help to identify spammers [73]. A very important fact is that spammers are *reactive*, namely they actively oppose every successful anti-spam effort [29], so that performance of a new method usually decreases after its deployment. Pu and Webb [74] analyze the evolution of spamming techniques, showing that methods of constructing spam become extinct if filters are effective enough to cope with them or if other successful efforts are taken against them. A study of network-level behavior of spammers by Ramachandran and Feamster [75] showed that the majority of spam comes from a few concentrated parts of IP address space, and that a small subset of sophisticated spammers use temporary route announcements in order to remain untraceable.

2.2 Anti-Spam Legislation Efforts

The huge and various damage caused by spam, including financial loss and violation of laws by broadcasting prohibited materials, resulted in the need for a legislative response. Noticeable efforts in this field are EU Privacy and Electronic Communications Directive, and US CAN-SPAM Act.

The European Parliament passed the Privacy and Electronic Communications Directive 2002/58/EC in July 2002. The directive prohibits unsolicited commercial communication unless "prior explicit consent of the recipients is obtained before such communications are addressed to them". An overview of the directive is given by Lugaresi [61]. In case of Italy, in particular, Section 130 of "Personal Data Protection Code" (Legislative Decree no. 196 of 30 June 2003) states that "the use of automated calling systems without human intervention for the purposes of direct marketing or sending advertising materials, or else for carrying out market surveys or interactive business communication shall only be allowed with the users consent".

US CAN-SPAM Act (Controlling the Assault of Non-Solicited Pornography and Marketing Act) of 2003 allows unsolicited commercial email, but places several restrictions on it. In particular, it demands to include a physical address of the advertiser and an opt-out link in each message, to use legitimate return email address, and to mark the messages clearly as advertisements, and prohibits to use descriptive subject lines, to falsify header information, to harvest email addresses on the Web, and to use illegally captured third-party computers to relay the messages. Grimes [42] shows, that the actual compliance with the CAN-SPAM act was low from the very beginning and became even lower in the following years, being equal to about 5.7% in 2006.

For more information on this topic, one may refer to an analysis of the EU and the US anti-spam legislation by Moustakas et al. [68], and to an overview of anti-spam legislation of differen countries prepared by the International Telecommunication Union [47].

2.3 Modifying Email Transmission Protocols

One of the proposed ways of stopping spam is to enhance or even substitute the existing standards of email transmission by new, spam-proof variants. The main drawback of the commonly used Simple Mail Transfer Protocol (SMTP) is that it provides no reli-

able mechanism of checking the identity of the message source. Overcoming this disadvantage, namely providing better ways of sender identification, is the common goal of Sender Policy Framework (SPF, formerly interpreted as Sender Permitted From) [92], Designated Mailers Protocol (DMP) [30], Trusted E-Mail Open Standard (TEOS) [82], and SenderID (sometimes also spelled Sender ID) [85]. A comparison and discussion of this kind of proposals is given by Levine and DeKok [57]. SenderID, being released in 2004, has grown quite popular already. According to Goodman et al. [39], almost 40% of legitimate email is today SenderID-compliant. The principle of its work is the following: the owner of a domain publishes the list of authorized outbound mail servers, thus allowing recipients to check, whether a message which pretends to come from this domain really originates from there. A discussion of the problem of fake IP addresses in email messages and ways of overcoming it by changes in standards is given by Goodman [36].

The idea underlying another group of proposals to amend the existing protocols is to add a step to the mail sending process that represents a minor obstacle for sending few emails, but a major one for sending great number of messages. Efforts in this direction were made already in 1992 [28], when it was proposed to ask sender to compute a moderately hard function before granting him the permission to sent a message. Another proposal [84] was to establish a small payment for sending an email message, neglible for a common user, but big enough to prevent a spammer to broadcast millions of messages. An interesting version of this approach is Zmail protocol [51], where a small fee is paid by the sender to the receiver, so that a common user who sends and receives nearly equal amount of messages gets neither damage no profit from using email, while spamming becomes a costly operation. Another approach is to use simple tests that allow the system to distinguish human senders from robots [12], for example to ask the user to answer a moderately easy question before sending the message. One disadvantage of this approach is that such protection is annoying to human senders. Duan et al. [27] propose to use a differentiated email delivery architecture to handle messages from different classes of senders in different ways. For example, for some classes messages are kept on the sender's mail server until the receiver asks to transmit them to him.

2.4 Local Changes in Email Transmission Process

Some solutions do not require global protocol changes but propose to manage email in a different way locally. Li et al. [59] and Saito [78] propose slowing down the operations with messages that are likely to be spam. A similar idea is discussed in the technical report by Twining et al. [91], who propose to use the past behavior of senders for fast prediction of message category, and then process supposed spam in a lower priority queue and supposed legitimate mail in a higher priority queue. In this way the delivery of legitimate mail is guaranteed, but it becomes hard to broadcast many spam messages at once. Yamai et al. [98] pointed out that when a spammer falsifies the sender identity in the messages, the server corresponding to the falsified address receives a great number of error mails. Yamai and collaborators propose to solve this problem by using a separate mail transfer agent for the error messages. Goodman and Rounthwaite [37] point to the possibility of controlling not only ingoing, but also outgoing spam, stopping it on the level of email service provider used by a spammer.

3 Learning-Based Methods of Spam Filtering

Filtering is a popular solution to the problem of spam. It can be defined as automatic classification of messages into spam and legitimate mail. Existing filtering algorithms are quite effective, often showing accuracy of above 90% during the experimental evaluation (see, for example, the evaluation performed by Lai and Tsai [53]). It is possible to apply the spam filtering algorithms on different phases of email transmission: at routers (see for example the paper by Agrawal et al. [1]), at the destination mail server, or in the destination mailbox. It must be mentioned

that filtering on the destination point solves the problems caused by spam only partially: a filter prevents end-users from wasting their time on junk messages, but it does not prevent resources misuse, because all the messages are delivered nevertheless.

In general, a spam filter is an application which implements a function:

$$f(m,\theta) = \begin{cases} c_{spam}, \textit{ if the decision is ``spam''} \\ c_{leg}, \textit{ otherwise} \end{cases}$$

where m is a message to be classified, θ is a vector of parameters, and c_{spam} and c_{leg} are labels assigned to the messages.

Most of the spam filters are based on machine learning classification techniques. In a learning-based technique the vector of parameters θ is the result of training the classifier on a pre-collected dataset:

$$\theta = \Theta(M),$$
$$M = \{(m_1, y_1), ...(m_n, y_n)\},\ y_i \in \{c_{spam}, c_{leg}\},$$

where $m_1, m_2, ...m_n$ are previously collected messages, $y_1, y_2 ... y_n$ are the corresponding labels, and Θ is the training function.

According to Fawcett [29], the following peculiarities of spam filtering task cause problems from the point of view of data mining: skewed class distribution (the proportion of spam to legitimate mail varies greatly), unequal and uncertain error costs, disjunctive and changing target concept (the content of spam changes with time), and reactive adversaries. Another problem is the need for sufficient amount of training data. Addressing this issue, Chan et al. [14] proposed to use semi-supervised learning, namely a technique called *co-training*, for spam filtering. This technique allows the learner to start off with a small amount of labeled training data, which is used for initial training of the classifier, and a larger amount of unlabeled training data, which is then labeled in an iterative process and used to train the classifier better.

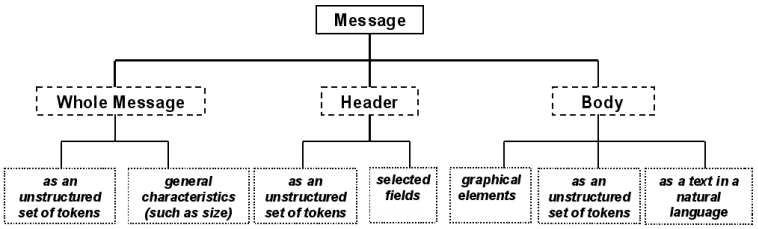
For all the algorithms of email classification there exists the problem of finding a reasonable trade-off between two types of errors: classifying legitimate mail as spam and classifying spam as legitimate mail. While classifying several spam messages as legitimate mail just annoys the user, the opposite situation may lead to the actual loss of valuable information. A solution for finding a trade-off based on game theory is proposed by Androutsopoulos et al. [7]. Also, Yih et al. [100] propose and discuss two techniques of training filters with low false positive rates. Nevertheless, we must remember, that different users have different requests, so it is reasonable to consider the relative cost of the two types of errors as a user-defined parameter [66].

The development of a new filter can be simplified by some existing software tools. Here we can mention Spamato system [2] that provides a uniform user-friendly software framework for spam filtering algorithms in order to simplify practical implementation of new filters, and the Email Mining Toolkit (EMT) [44], a data mining toolkit designed to analyze offline email corpora.

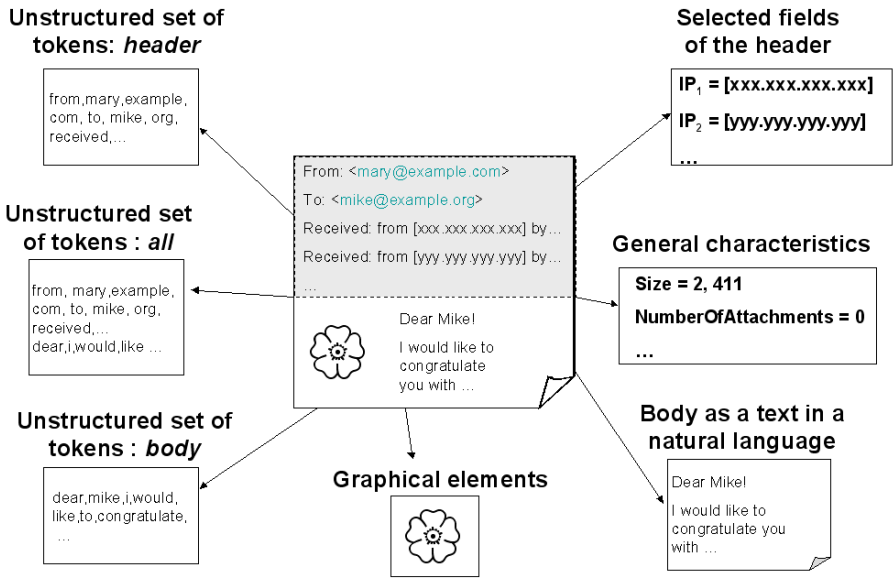
3.1 What to Analyze?

In order to classify new messages, a spam filter can analyze them either separately (for example, just checking the presence of certain words in case of keyword filtering) or in groups (for example, a filter may consider that arrival of a dozen of substantially identical messages in five minutes is more suspicious then arrival of one message with the same content). In addition to this, a learning-based filter analyzes a collection of labeled training data (pre-collected messages with reliable judgements), and a filter which involves user collaboration receives also multiple user judgements about some of the new messages for the analysis.

An email message consists of two parts, namely body and header. Message body is usually a text in a natural language, possibly with HTML markup and graphical elements. Header is a structured set of fields, each having name, value, and specific meaning. Some of this fields, like *From*, *To*, or *Subject*, are standard, and others may depend on the software involved in message transmission, such as spam filters installed on mail servers. *Subject* field contains what the user sees as the subject of the message and is often treated as a part of the message body. The body is



(a) Taxonomy.



(b) Example.

Figure 1: What to analyze? Message structure from the point of view of feature selection.

sometimes referred to as the content of the message. We must mention that non-content features are not limited to the features of the header. For example, a filter may consider the message size as a feature [44].

For each method of message analysis its designer must choose a way of doing feature selection, namely decide what parts of the messages are relevant for the analysis. The simplest way of doing feature selection is the 'bag of words' model, which represents the message as an unstructured set of tokens, namely sequences of chracters separated by spaces and/or punctuation marks. This model can be used to characterize any part of a message, or a message as a whole. In this case, presence of a certain word in the message is considered a binary feature of the message. A somewhat more sophisticated approach is to consider the occurences of same word in different parts of the message (say, 'John' in the message body and 'John' in the 'From' field) as different features. This approach, though makes some use of the message structure, does not really exploit the differences between text in the body and technical information in the header, so further in the discussion we will make no difference between this approach and the plain 'bag of words'. Also a weighted variant can be used, when the features are not binary, but reflect the importance of the token in some way, for example the number of occurrences of the token in the message can be used as the weight of this token. It is possible to use all the features, or to select top N features by some measure. Zhang et al. [102] name three measures that can be used to order the features: document frequency, information gain, and χ^2 (the definitions are given in Table 1).

Natural language processing provides some alternative ways of selecting features from the body. The most simple way is enhancing the 'bag of words' model with stemming (removing affixes) and/or stopping (ignoring the most frequent words). For the message header analysis, more sophisticated ways of selecting features take the header structure into account, extracting only some special kind of information. Yeh et al. [99] propose a complex approach based on meta-heuristics, using knowledge about typical behaviors of spammers to specify features for recognizing spam (for example the "From" field empty or missing, or the date illegal or very old, are considered signs of spam message). Hershkop [44] uses a wide range of non-content features, including features extracted from the header, such as sender and recipient email names, domain names and zones, and general characteristics of the message, such as the message size and the number of attachments.

3.1.1 Feature extraction for image-based filtering.

Apart from text, a message can also contain graphical images. After the distribution of content-based filtering techniques, the spammers adopted the use of image spam. The text of an advertizement is placed in an image, so that it is impossible to analyze the message content with plain text-based filters. This led to the need for filters based on image analysis. In image-based filtering the main issue is to find features both relevant and easy to extract, while the classification itself can be further perfomed by state-of-the-art algorithms.

The fully-functional optical character recognition (OCR) procedure is computationally expensive, so usually simplified models are proposed to recognize spam in images. In particular, Aradhye et al. [8] extract five features from the images, namely the fraction of the image occupied by regions identified as text, and color saturation and color heterogeneity calculated separately for text and non-text regions. A similar approach to feature extraction for image-based filtering was proposed by Wu et al. [97]. In addition to detecting the size and the number embedded text regions without actual text recognition, they characterize a banner as a special kind of image (very narrow in width or height, and with a large aspect ratio), and use the number of banner-like images as an additional feature. Lately, Dredze et al. [25] introduced a new approach, which relies only on features which take very small time to extract, avoiding not only OCR, but in general any computations more complicated than simple edge detection. Thus, the features used in this work are selected among those that do not require image analysis at all (for example, file format, height and width of the image, or file size), and those that are retrieved through very

Measure	Formula
Document frequency	$\lvert\{m_j \lvert m_j \in M \text{ and } f_i \text{ occurs in } m_j\}\rvert$
Information gain	$\sum_{c\in\{c_{spam},c_{leg}\}} \left(\sum_{f\in\{f_i,\neg f_i\}} \hat{P}(f,c) \log \frac{\hat{P}(f,c)}{\hat{P}(f)\cdot\hat{P}(c)} \right)$
χ^2	$\frac{\lvert M\rvert \cdot [\hat{P}(f_i,c_{spam})\cdot\hat{P}(\neg f_i,c_{leg}) - \hat{P}(f_i,c_{leg})\cdot\hat{P}(\neg f_i,c_{spam})]^2}{\hat{P}(f_i)\cdot\hat{P}(\neg f_i)\cdot\hat{P}(c_{spam})\cdot\hat{P}(c_{leg})}$

Table 1: Measures of feature relevance used for ordering features. Each measure applies to a feature. M is the set of all training messages, c_{spam} and c_{leg} are the labels of spam class and legitimate mail class correspondingly, f_i is a binary feature (for example "the word *free* is present in the message"), and $\neg f_i$ is the negation of the feature f_i (for example "the word *free* is NOT present in the message"). All the probabilities are estimated with frequencies.

simple analysis of images (for example, average color or color saturation). Similarly, Wang et al. [94] use such fast-to-extract features as color histogram, orientation histograms, and coefficients of wavelet transformation of the image. All this methods showed reasonably high accuracy, but, as explicitly stated by Dredze et al. [25], such approaches are vulnerable to reactivity. It can be well seen on the example of features used to characterize banners, which can obviously be easily avoided by spammers and already today are unlikely to be helpful.

Despite the general desire to avoid OCR for the reasons of low speed, Fumera et al. [32] note that it may be reasonable to apply OCR-based recognition in the rare cases when simpler filters are unable to provide a confident decision. They show that application of state-of-the-art text categorization techniques to the text extracted from the images can be quite efficient. Providing positive results, they nevertheless observe that the spammers can easily react by applying techniques which will pose problems to OCR without decreasing human readability of text – ironically, the same techniques which are used in the tests designed to distinguish human senders from robots.

3.2 How to Analyze?

The first filters were based plainly on checking presence of certain predefined tokens in the message body (keyword filtering) or in the information about the sender (blacklist/whitelist filtering). Though this approches are not themselves learning-based, it is necessary to mention them in the beginning of this section, because a great number of later filters are in fact sophisticated improvements of the same two initial ideas. While keword filtering was completely replaced by its learning-based descendants (primarly Naïve Bayes), blacklists and whitelists are used until now as parts of more complex anti-spam solutions [66]; apart from personal blacklists, the public up-to-date registers of known spammers exist (see for example [49]) and are widely used. One more related method is greylisting [43], when a message which is neither in the whitelist nor in the blacklist is temporarily rejected; if an attempt of transmission on the same message is held later, the message is accepted. This method rests on the assumption that spammers do not always retry sending their messages, and those who do will probably be listed in public blacklists during the time gap between the two attempts.

Below we provide short descriptions of the existing filtering methods.

3.2.1 Methods Based on Bag-of-Words Feature Extraction

Learning-based spam filters that treat the input data as an unstructured set of tokens, can be applied both to the whole message and to any part of it. For this group of filters we can state the problem as follows. Let there be two classes of messages: spam and legiti-

mate mail. Let us than have a set of labeled training messages, each message being a vector of d binary features and each label being c_{spam} or c_{leg} depending on the class of the message. Thus, the training data set M, once pre-processed in this way, can be described as:

$$X = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), ..., (\bar{x}_n, y_n)\},$$
$$\bar{x}_i \in \mathbb{Z}_2^d, y_i \in \{c_{spam}, c_{leg}\},$$

where d is the number of features used. Then, given a new sample $\bar{x} \in \mathbb{Z}_2^d$ the classifier should provide a decision $y \in \{c_{spam}, c_{leg}\}$.

Naïve Bayes. In 1998 the Naïve Bayes classifier was proposed for spam recognition [71, 77]. It became widely known and used due to Paul Graham's popular article "A Plan for Spam" [40]. This classifier, when applied to text, can be considered an improved learning-based variant of keyword filtering. It rests on the so-called naive independence assumption, namely that all the features are statistically independent. The basic decision rule can be defined as follows:

$$f(\bar{x}) = \underset{y=\{c_{spam}, c_{leg}\}}{\operatorname{argmax}} \left(\hat{P}(y) \prod_{j:x^j=1} \hat{P}(x^j = 1|y) \right),$$

where x^j is the jth component of the vector $\bar{x}$, $\hat{P}(y)$ and $\hat{P}(x^j = 1|y)$ are probabilities estimated using the training data. Several variants of Naïve Bayes were applied to spam filtering, an overview and comparison of them can be found in the article by Metsis et al. [65]. Though the classifier is very fast as it is, Li and Zhong [58] proposed to make it even faster by using approximate classification techniques. Their version of the algorithm achieves significant increase in speed without loosing much in accuracy.

k-Nearest Neighbor. The k-Nearest Neighbor (k-NN) classifier was proposed for spam filtering by Androutsopoulos et al. [5]. With this classifies the decision is made as follows: k nearest training samples are selected using a predefined similarity function, and then the message $\bar{x}$ is labeled as belonging to the same class as the majority among this k samples.

Support Vector Machines. Another classifier proposed for spam filtering is Support Vector Machine (SVM) [26]. Given the training samples and a predefined transformation $\Phi : \mathbb{R}^d \to F$, which maps the features to a transformed feature space, the classifier separates the samples of the two classes with a hyperplane in the transformed feature space, building a decision rule of the following form:

$$f(\bar{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\bar{x}_i, \bar{x}) + b \right),$$

where $K(\bar{u}, \bar{v}) = \Phi(\bar{u}) \cdot \Phi(\bar{v})$ is the kernel function and $\alpha_i, i = 1..n$ and b maximize the margin of the separating hyperplane. The value -1 corresponds to c_{leg}, 1 corresponds to c_{spam}. SVM was proposed in particular to classify the vectors of features extracted from images [8].

Lately two improvements of this method of filtering appeared. Sculley and Wachman [83] proposed a version of SVM, called Relaxed Online SVM, which reduces greatly the computational cost of updating the hypothesis, in particular by training only on actual errors. Blanzieri and Bryl [9] presented an SVM-based filtering algorithm which improves the accuracy by using locality in the spam phenomenon.

Term Frequency-Inverse Document Frequency. The name Term Frequency-Inverse Document Frequency (TF-IDF) actually applies to a term-weighting scheme, which is defined as follows:

$$w_{ij} = tf_{ij} \cdot \log \frac{n}{df_i},$$

where w_{ij} is the weight of ith term (token) in the jth document (message), tf_{ij} is the number of occurrences of the ith term in the jth document, df_i is the number of messages in which the ith term occurs, and n, as above, is the total number of documents in the training set. This scheme can be combined with the Rocchio algorithm, a detailed description of which can be found in the paper by Joachims [48]. Such combination results in a quite accurate classifier [26], which is sometimes also referred to as TF-IDF in the literature.

Boosting. Boosting is a general name for the algorithms based on the idea of combining many hypotheses (for example one-level decision trees). At each stage of the classification procedure a weak (not very

accurate) learner is trained, and its output is used to reweight the data for the future stages: greater weight is assigned to the samples which are misclassified. For spam filtering boosting was proposed by Carreras and Márquez [13].

3.2.2 Language-based filters

Another group of methods uses the fact that the message body is a text in a natural language. We must mention that methods discussed in this section can in practice be applied also to message headers or whole messages, however the motivation proposed in the literature for their application on spam filtering relies on the fact that they are effective in natural language text classification. In fact, the same motivation can as well be appled to the methods based on compression models, namely dynamic Markov compression and prediction by partial matching, which were nevertheless successfuly used with the data extracted from both bodies and headers of the messages [11].

Chi by degrees of freedom. This method, which is usually used for document authorship identification, is proposed for spam filtering by O'Brien and Vogel [70]. Messages are represented in terms of character or word N-grams. The idea of the method is to compare the similarity of a new message to the labeled messages using the chi-by-degrees-of-freedom test, which is calculated by dividing the value of the χ^2 test by the number of degrees of freedom.

Smoothed N-gram language models. Medlock [64] used smoothed higher-order N-gram models. N-gram language models are based on the assumption that the existence of a certain word at a certain position in a sequence depends only of the previous $N-1$ words.

3.2.3 Filters based on non-content features

The methods based on structured analysis of the header and of meta-level features, such as number of attachments, use specific technical aspects of email and so they are specific to spam filtering.

Analyzing SMTP path. Leiba et al. [56] present a filtering method based on analyzing IP addresses in the reverse-path and ascribing reputation to them according to amount of spam and legitimate mail delivered through them. Both this and the subsequent method can be viewed as development of the idea of blacklisting and whitelisting.

Analyzing the user's social network. The algorithm proposed by Boykin and Roychowdhury [10] analyzes 'From', 'To', 'Cc' and 'Bcc' fields of the message headers in order to build a graph of social relations of the user, and then uses this graph in order to classify new messages. The idea of extracting the user's social network from his mailbox was further developed by Chirita et al. [15] and by Golbeck and Hendler [34].

Analyzing behaviors. Behavior-based filtering rests on extracting knowledge about the behavior behind a given message or group of messages from their non-content features, and comparing it to predefined or extracted knowledge about the typical behaviors of malicious and normal users. Examples are the works of Yeh et al. [99], and Hershkop [44], both already mentioned in Section 3.1. Yeh et al. [99] use well-known behaviors of spammers, such as using incorrect dates. Hershkop [44] proposes a number of behavior models, among them recipient frequency and histograms of user's past activity, that are based on non-content features and can be used to detect spam and viruses as anomalies in the email flow.

3.2.4 Collaborative spam filtering

Certain efforts are made to achieve better spam filtering through the collaboration of users. The usual way of such collaboration is sharing the knowledge about spam between P2P users [54, 104], or gathering spam reports from the users on a mail server (like in Google's Gmail[1]). In such situation of data exchange between users the issue of privacy arises. Damiani et al. [22] propose a privacy-preserving approach to P2P spam filtering system. In particular, spam reports in their system are sent without indicating the user who is the source of the report. Mo et al. [67] propose a multi-agent system for collaborative spam filtering, in which each message is first

[1] http://gmail.google.com/

classified as spam, legitimate mail or suspicious mail by a local agent, and only for suspicious messages the collaborative judgement is requested. While usually the users are proposed to exchange opinions or information about emails, Garg et al. [33] propose to exchange trained filters instead, thus significantly reducing the amount of data transmitted. Another interesting effort for collaborative spam fighting is Project Honey Pot [45], intended to identify email address harvesters with the help of specially generated email addresses.

3.2.5 Hybrid approaches

We must mention that it is also possible to combine different algorithms, especially if they use unrelated features to produce a solution [56, 102].

3.2.6 Overview of the methods

In Table 2 we give a wide list of the spam filtering algorithms proposed in the literature. In the same cell of the table we group similar algorithms that are based on the same idea but may have some differences. For example, Drucker et al. [26] use C4.5 decision trees as a weak learner for boosting algorithm, and Androutsopoulos et al. [6] use regression stumps. Here we refer only to the articles directly related to spam filtering, but many of the listed methods were known and used for other tasks before. In particular we must mention that RIPPER and TF-IDF classifiers were applied to the similar task of email classification by topic as early as 1996 [17].

3.3 Opposing Reactivity

The methods of spamming are improving together with the methods of spam filtering. Spammers try to attack filters, namely to decrease filtering effectiveness. Following the systematization proposed by Wittel and Wu [95] we can categorize attacks on spam filters in the following way:

- **Tokenization attacks**, when the spammer intends to prevent correct tokenization of the message by splitting or modifying features, for example putting extra spaces in the middle of the words.
- **Obfuscation attacks**, when the content of the message is obscured from the filter, for example by means of encoding.
- **Statistical attacks**, when the spammer intends to skew the message's statistics. If the data used for a statistical attack is purely random, the attack is called *weak*; otherwise it is called *strong*. An example of strong statistical attack is *good word attack* [60].

The reactivity of spammers requires countermeasures from filter developers, so in the field of spam filtering a direction appeared which we may call *opposing reactivity*. For example, a popular trick of spammers is to misspell the most 'spam-like' words, for example writing 'vi@gra' instead of 'viagra'. A way to solve this problem using hidden Markov model is proposed by Lee and Ng [55]. Also we can mention that the whole issue of image spam initially arose as a part of the problem of reactivity, and so the image-based spam filtering as such can be considered opposition to reactivity.

4 Commercial and Non-Commercial Software Solutions

Spam filtering is not only a subject of scientific research, but also a wide and well-established field of software development. Available commercial and non-commercial solutions combine different techniques of message filtering. Moreover, they use protocol extensions and are sometimes integrated into single software solutions with anti-virus protection. An overview of some products is given in Table 3. The meanings of the column titles are as follows:

- **Whitelists/blacklists**: use of various personal and public blacklists and whitelists;
- **Managing replies**: using additional mechanisms to ensure that replies to the user's messages are not classified as spam;

Method	**Can be applied to**	**Applied to**	**Used in**
RIPPER	B,H,W	B	[26]
Stacking	B,H,W	B	[79, 105]
Naïve Bayes	B,H,W	B,H,W	[5, 4, 3, 6, 14, 41, 53, 62, 71, 77, 102, 105]
Flexible Bayes	B,H,W	B	[6]
Boosting	B,H,W	B,H,W	[6, 13, 26, 102, 105]
Maximum Entropy Model	B,H,W	B,H,W	[101, 102]
Support Vector Machines	B,H,W	B,H,W	[6, 9, 14, 26, 52, 53, 83, 96, 102, 105]
k-NN	B,H,W	B,H,W	[5, 23, 53, 80, 102, 105]
Centroid-based	B,H,W	B	[87]
TF-IDF	B,H,W	B,H,W	[53, 26]
Pattern discovery	B,H,W	B	[76]
Self-organizing Feature Maps (SOM)	B,H,W	B	[62]
Learning Vector Quantization (LVQ)	B,H,W	B	[16]
Committee Machines	B,H,W	B	[107]
Compression Models	B,H,W	B,W	[11]
Clustering	B,H,W	B	[81]
Rough Set Based Model	B,H,W	B	[103]
χ By Degrees Of Freedom	B	B	[70]
Smoothed N-gram Modelling	B	B	[64]
SMTP-path Analysis	H	H	[56]
Social Networks	H	H	[10, 15]

Table 2: Spam Filtering Algorithms. The following abbreviations are used: B - body, H - header, W - whole message.

Product	**Whitelists/blacklists**	**Managing replies**	**Using decoy accounts**	**Protocol extensions**	**Anti-virus/anti-spyware**	**User collaboration**	**Message analysis**	**Bayesian**	**Image analysis**	**Downloading updates**	**Price**
Server-side software solutions											
Symantec Mail Security for SMTP	+		+		+		+			+	Not stated on the site
MailCleaner	+				+		+	+	+	+	Complex sys. of prices
Solutions suitable both for client and server side											
SpamAssassin	+						+	+			Free
Bogofilter							+	+			Free
Client-side software solutions											
CA Anti-Spam	+						+			+	€39.95
Vanquish vqME	+	+		+			+				$34.95/year
Cloudmark Desktop						+					$39.95
Allume SpamCatcher						+	+			+	$29.99
MailWasher Pro	+						+				$37
POPFile							+	+			Free
Spamihilator	+					+	+	+			Free
SpamPal	+										Free
K9	+						+	+			Free
G-Lock SpamCombat	+						+	+			Free
Software solutions supplied with a hardware base											
BorderWare Email Security Gateway	+			+	+		+		+	+	Not stated on the site
Barracuda Spam Firewall				+	+		+		+	+	Complex sys. of prices

Table 3: Methods used in some software anti-spam solutions. The meanings of the column titles are explained in Section 4. The addresses of websites are given in Table 4.

Product	Website address
Symantec Mail Security for SMTP	http://www.symantec.com/enterprise/products/overview.jsp?pvid=845_1
MailCleaner	http://www.mailcleaner.net/
SpamAssassin	http://spamassassin.apache.org/
Bogofilter	http://bogofilter.sourceforge.net/
CA Anti-Spam	http://home3.ca.com/STContent/landingpages/Products/Antispam/ASPM001/index.aspx
Vanquish vqME	https://www.vqme.com/
Cloudmark Desktop	http://cloudmark.com/desktop/
Allume SpamCatcher	http://www.allume.com/win/spamcatcher/
MailWasher Pro	http://www.mailwasher.net/
POPFile	http://popfile.sourceforge.net/
Spamihilator	http://www.spamihilator.com/
SpamPal	http://www.spampal.org/
K9	http://keir.net/k9.html
G-Lock SpamCombat	http://www.glocksoft.com/sc/
BorderWare Email Security Gateway	http://www.borderware.com/products/email-security-gateway/
Barracuda Spam Firewall	http://www.barracudanetworks.com/ns/products/spam_overview.php

Table 4: Addresses of the official websites of the products presented in Table 3.

- **Using decoy accounts**: collecting spam messages on decoy accounts for future extraction of fingerprints or rules;
- **Protocol extensions**: support of protocol extensions intended to prevent falsifying the sender's identity or to ensure that a message is legitimate by asking the sender for confirmation;
- **Anti-virus/anti-spyware** : integrating an anti-virus and/or anti-spyware solution into the same product;
- **User collaboration**: support of sharing data about spam among the users of the product;
- **Message analysis**: methods of filtering more sophisticated than blacklisting and whitelisting;
- **Bayesian**: Bayesian algorithm is used for message analysis, probably in combination with other techniques;
- **Image analysis**: use of algorithms of analysis of graphical content;
- **Downloading updates**: the product regularly downloads updates for its database from a server;
- **Price**: the price of the product as given on the official site, as of May, 2007.

The table is based only on the explicit statements on the official websites of the products, and thus may be incomplete. It does not provide real performance comparison and is not intended to advice any choice between this products, but rather to show which techniques are used in practical solutions. We do not include the information about the effectiveness of the solutions into the table, because it is stated only for few products, and sometimes the accuracy is claimed to be 100%, which seems rather a marketing slogan than a piece of information that can be used for comparison.

We can see that practical solutions often combine various ways of blacklisting and whitelisting with more complex filtering methods. An interesting point is that many products use Bayesian filtering. The reason for this is probably the following: approaches based on Naïve Bayes, though shown by many stud-

ies to be slightly outperformed by other techniques, have the advantage of being very fast and fit for continuous on-line training.

5 Method Evaluation and Comparison

The great number and variety of spam filtering methods results in the need for evaluation and comparison of them. The usual way of testing a filter is applying it to a corpus of previously gathered mail messages sorted into spam and legitimate mail. The most simple measure used to express the results of such testing is filtering accuracy, namely percentage of messages classified correctly [53], which has the disadvantage of making no difference between false positives and false negatives. More informative measures are spam recall and spam precision. Androutsopoulos et al. [4] propose to use the relational cost λ of the two types of errors as a variable parameter, and introduce several new measures based on it: weighted accuracy, weighted error rate, and a total cost ratio (TCR). TCR is the relative cost of using the filter (and so having some false positives and some false negatives) to using no filter at all (and so having all the spam misclassified, but all the legitimate mail classified correctly). Table 5 gives the formulae of the measures named above. It is also possible to test a filter in real-life conditions. A straightforward way is to use it on one's mailbox or mail server. Nevertheless, such testing, having the advantage of using up-to-date data, is more time-consuming (Michelakis et al. [66] chose a period of seven months to test their filter). Usually a previously known method is tested simultaneously in the same way to provide a quality baseline. The Naïve Bayes classifier is often chosen for this purpose. However, Naïve Bayes has already been shown to be outperformed by many other methods (see for example [13, 102, 16]), so now a more accurate baseline method is needed, for example Support Vector Machines, as done by Sasaki and Shinnou [81].

Some mail corpora are made publicly available by their editors. The list of public corpora is given in Table 7. The properties of spam change with time, so the older is a corpus, the less the results can be accepted as an estimation of present real-world performance. We must mention here that the LingSpam corpus, being rather old, is still actively used, and this may lead to out-of-date performance results. Creation of new public corpora is slowed down by privacy issues: people are certainly unwilling to publish their private email. For this reason some studies use either corpora that are not publicly available [56, 99], or both private and public corpora [18, 53]. One of the largest public sources of legitimate mail for experiments, the so-called Enron Corpus[2] [50], was made available during the legal investigation. The data from this repository was later included in the Spam Track 2005 corpus and Enron-Spam corpora. Being against publishing their legitimate mail, people usually do not object publishing spam from their mailboxes, so it is possible to collect a really large repository of pure spam. For example, SpamArchive project proposes over 220,000 spam messages for experimental needs.

Some studies are dedicated to comparison of more than two filters [6, 26, 53, 102]. In particular, Lai and Tsai [53] make a complex comparison of four different methods (Naïve Bayes, SVM, k-nearest neighbor, and TF-IDF) applied to different parts of a message and show that, at least on their corpora, analyzing the header usually gives better results than analyzing the body or the whole message. According to the results presented by Zhang et al. [102], the highest TCR is achieved by using both headers and bodies, but using header alone again leads to better results than using body alone. A comparison of 44 spam filters supplied by 12 groups of developers was performed on Spam Track[3] on the Text Retrieval Conference (TREC) in 2005. According to the final report [18], the best performance was shown by one of the filters supplied by Jožef Stefan Institute and based on compression models [11], able to achieve spam misclassification rate of 1.17% with false positive rate of 0.1%. Another method which showed high resuls was gradient descent of a logistic regression model [38]. The method of testing used in this competition is different from

[2] Available at http://www-2.cs.cmu.edu/~enron/

[3] http://plg.uwaterloo.ca/~gvcormac/spam/

Measure	Formula
Accuracy	$\frac{n_{L\to L}+n_{S\to S}}{n_{L\to L}+n_{L\to S}+n_{S\to L}+n_{S\to S}}$
Error rate	$\frac{n_{L\to S}+n_{S\to L}}{n_{L\to L}+n_{L\to S}+n_{S\to L}+n_{S\to S}}$
False positive rate	$\frac{n_{L\to S}}{n_{L\to L}+n_{L\to S}}$
Spam recall	$\frac{n_{S\to S}}{n_{S\to L}+n_{S\to S}}$
Spam precision	$\frac{n_{S\to S}}{n_{L\to S}+n_{S\to S}}$
Weighted accuracy	$\frac{\lambda\cdot n_{L\to L}+n_{S\to S}}{\lambda\cdot(n_{L\to L}+n_{L\to S})+n_{S\to L}+n_{S\to S}}$
Weighted error rate	$\frac{\lambda\cdot n_{L\to S}+n_{S\to L}}{\lambda\cdot(n_{L\to L}+n_{L\to S})+n_{S\to L}+n_{S\to S}}$
Total cost ratio	$\frac{n_{S\to L}+n_{S\to S}}{\lambda\cdot n_{L\to S}+n_{S\to L}}$
ROC curve	*True positive rate plotted against false positive rate*

Table 5: Measures of filtering performance. Following Androutsopoulos et al. [4], $n_{L\to L}$ and $n_{S\to S}$ are the numbers of legitimate and spam messages classified correctly, $n_{L\to S}$ and $n_{S\to L}$ are the numbers of legitimate and spam messages misclassified, and λ is the relative cost of the two types of errors.

Corpus	Available At
PU1, PU2, PU3, PUA, LingSpam Enron-Spam datasets (Enron1, Enron2, Enron3, Enron4, Enron5, Enron6)	http://www.aueb.gr/users/ion/publications.html
Spamassassin	http://spamassassin.apache.org/publiccorpus/
ZH1 Chinese	http://homepages.inf.ed.ac.uk/s0450736/spam/
GenSpam	http://www.cl.cam.ac.uk/users/bwm23/
Spam Track corpus	http://plg.uwaterloo.ca/~gvcormac/spam/
Spambase	http://www.ics.uci.edu/~mlearn/MLSummary.html
SpamArchive	http://www.spamarchive.org/

Table 6: Public Data Repositories.

Corpus	Number of messages	Spam rate	Headers included	Encrypted	Year of creation	Used in
PU1	1,099	44%	NO	YES	2000	[3, 6, 11, 102]
PU2	721	20%	NO	YES	2003	[6]
PU3	4,139	44%	NO	YES	2003	[6, 11]
PUA	1,142	50%	NO	YES	2003	[6]
LingSpam	2,893	17%	NO	NO	2000	[4, 11, 62, 79, 81, 102, 105, 107]
Spamassassin	6,047	31%	YES	NO	2002	[9, 11, 53, 16, 102]
ZH1 Chinese	1,633	74%	YES	YES	2004	[102]
GenSpam	41,404	78%	NO	NO	2005	[64]
Spam Track corpus	92,189	57%	YES	NO	2005	[11, 18, 38]
Enron1	5172	29%	NO	NO	2006	[65]
Enron2	5857	26%	NO	NO	2006	[65]
Enron3	5512	27%	NO	NO	2006	[65]
Enron4	6000	75%	NO	NO	2006	[65]
Enron5	5175	71%	NO	NO	2006	[65]
Enron6	6000	75%	NO	NO	2006	[65]
Spambase	4,601	39%	NO	YES	1999	[103]
SpamArchive	over 220,000	100%	YES	NO	-	

Table 7: Description of Public Data. 'YES' in the 'Encrypted' field means that tokens in the messages are encrypted to address personal privacy, or (in Spambase) only some extracted features of the messages are present in the corpus.

Id	Paper	Corpora used
A1	[5]	LingSpam
A2	[3]	PU1
A3	[6]	PU1, PU2, PU3 and PUA
Dr	[26]	Two specially created repositories
Ca	[13]	PU1
Ch	[16]	SpamAssassin
LT	[53]	SpamAssassin and a specially created repository
Le	[56]	Specially created repository
LZ	[62]	LingSpam
OV	[70]	Specially created repository
SS	[81]	LingSpam
So	[87]	Specially created repository
Z1	[102]	PU1, LingSpam, SpamAssassin and ZH1
ZZ	[103]	Spambase database
Z2	[105]	LingSpam
Zo	[107]	LingSpam

Table 8: Papers that present comparisons of two or more filtering techniques.

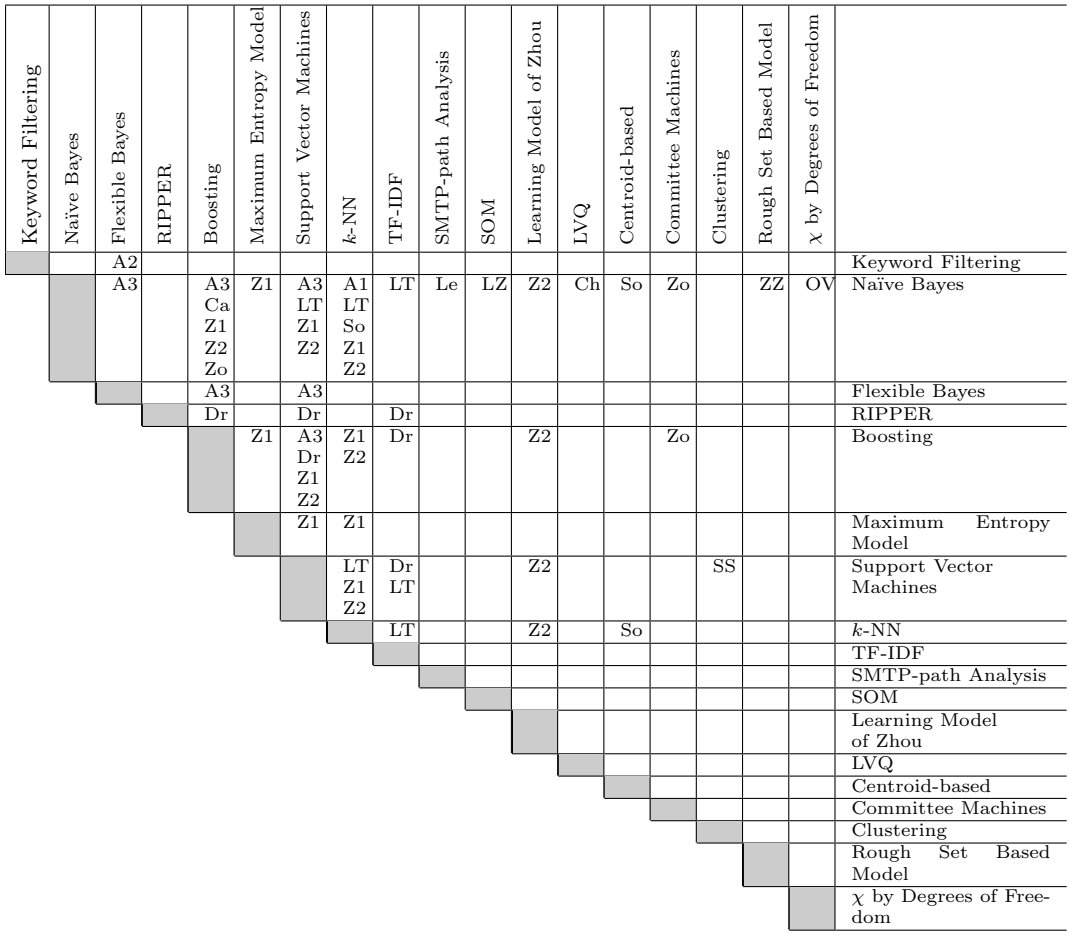


Table 9: Comparison of Spam Filtering Algorithms in the Literature. For references to the articles see table 8.

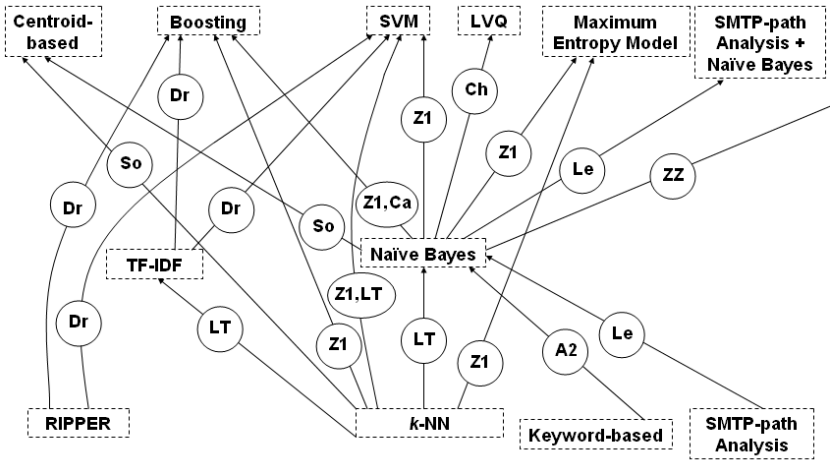


Figure 2: Graphical Comparison of Spam Filtering Algorithms in the Literature. An arrow from method A to method B with references on it means that A is outperformed by B according to the given article(s). An arrow is put only if there is an explicit claim on the relative performance of the two methods in the article. For references to the articles see table 8.

the usual one. Instead of commonly used offline testing, when the corpus is split into training and testing data, on-line testing is used: each message is first classified by the filter and then added to the training data. In this way the testing process emulates the real-life situation where the user corrects the errors made by the filter, so that the amount of training data gradually increases. Cormack and Bratko [20] discussed the differences between the testing approaches used in Spam Treck and other comparisons. They showed that, though there are important differences between batch and on-line evaluation, the methods which performed well on Spam Track also show good results being tested in a more conservative way. TREC Spam Filter Evaluation Tool Kit is available for download from the Spam Track website together with the data corpus. The approach used to create this corpus is described by Cormack and Lynam [19]. Competitions of spam filters were also arranged within TREC 2006[4], ECML/PKDD 2006[5], and CEAS 2007 conferences[6].

There is a a wide literature presenting comparison of small groups of filters, apart from the public competitions. In Table 8 we give a list of papers that present comparisons of two or more filtering techniques. In Table 9 we propose a systematization of comparisons of spam filtering methods presented in literature. Figure 2 represents the results of this comparisons. We must state here that accuracy and reliability of different comparisons presented in the tables may differ depending on data, ways of preprocessing, and peculiarities of methods of comparison. As a consequence, different comparisons cannot be combined in order to give some final judgement. For example, Leiba et al. [56] show that pure SMTP-path analysis is outperformed by Naïve Bayes on their repository, conversely Zhao and Zhang [103] show that Rough Set Based Model outperforms Naïve Bayes on the data from Spambase database. Obviously, this information is not enough to judge the relative performance of SMTP-path analysis and Rough Set Based Model.

Apart from the widely used accuracy measures, some other features are evaluated in different studies. Drucker et al. [26] and Zhou et al. [105] evaluate

[4]http://trec.nist.gov/pubs/trec15/t15_proceedings.html
[5]http://www.ecmlpkdd2006.org/challenge.html
[6]http://www.ceas.cc/2007/challenge/challenge.html

the classification speed. Boykin and Roychowdhury [10] analyze possible countermeasures that spammers may take to cheat the filter. Androutsopoulos et al. [4] evaluate the dependence of performance on training data size and attribute set size. For Spam Track, Cormack and Lynam [18] use learning curves to see how filter performance changes with time if the user retrains the filter continuously by correcting most of the classification errors.

6 Conclusion

In this paper we discussed the problem of spam and gave an overview of learning-based spam filtering techniques. There is no common definition of what spam is, but most of the sources agree that the core feature of the phenomenon is that spam messages are unsolicited. Spam causes a number of problems of both economical and ethical nature, which results in particular in the attempts of legislative definition and prohibition of spam. An important feature of the phenomenon of spam is the reactivity of spammers, in other words active intelligent opposition to every useful anti-spam technique. Another feature is the changeability of spam, which results partly from the reactivity of spammers, but also from changing content of the spam messages. One of the issues related to reactivity, namely falsification of the sender's identity, is fought by means of protocol extension. A serious obstacle for such approaches is that a new protocol must be willingly accepted by a great number of users to become really beneficial. At present at least one such solution, SenderID, has gained reasonable popularity, thus starting to influence the situation.

The most popular and well-developed approach to anti-spam is learning-based filtering. The current state of the art includes lots of filters based on various classification techniques applied to different parts of email messages. In the field of spam filtering the reactivity of spammers is noticeable, and attempts are made to predict and prevent the spammers' countermeasures. In general, local spam filtering has the drawback of solving the problem of spam only partially, because a filter saves user's time, but do not prevent resource misuse. The issue of changeability has no final solution yet, as it can be seen in particular from the necessity of frequent updates of databases in the commercial anti-spam software.

The great number of proposed filtering techniques causes the need for systematic evaluation and comparison. Efforts are made in this direction: evaluation methods and measures are proposed and repositories for testing are created, though the amount of experimental data publicly available is limited because of privacy issues. In the last years, the evaluation field became more systematic due to centralized contests of filters, such as the ones held within TREC, ECML/PKDD and CEAS conferences. Still, there exists no way to measure filter's stability against the reactivity of spammers. Apart from this, the increasing accuracy of the solutions will probably soon result in a situation where a big number of benchmark datasets will be required for real comparison of leading solutions.

From our overview of the field we can draw the following conclusions:

1. Spam filtering is quite effective, making the situation tolerable and thus probably being the cause of the slowness with which the useful protocol extensions are accepted by users. Because of the sufficient accuracy of the existing solutions, more attention is now given to narrower subtasks, such as analysis of image-based spam or coping with reactivity.

2. The reactivity of spammers is a major problem, and careful analysis of possible countermeasures is required for any new approach. The challenge to machine learning is to provide classification algorithms that are robust with respect to variation of the data that depends on classifier itself. As this ideal final goal seems to be unreachable as yet, in practice the providers of anti-spam techniques rather aim to be just *more reactive* then spammers, responding to new spamming techniques before they spread widely enough to change the balance.

A relevant issue is the influence of protocol-based and legislative approaches on the spam filtering problem. The increasing spread of SenderID gives hope

that the issue of falsifying the message source will soon be finally solved, thus limiting the range of methods of message obfuscation available to spammers and contributing to the accuracy of methods based on the analysis of the information contained in the header. The legislative approaches, in their turn, do not seem to influence the situation significantly, and no crucial improvement is likely to come in the near future.

In conclusion, we can say that the field of anti-spam protection is by now mature and well-developed. Then a question arises, why our inboxes are still often full of spam? Reactivity of spammers plays a role surely, but the countermeasures for their new tricks are proposed fast enough. So a possible answer is that we do not protect against spam in all the available ways. In other words, one point, which should always be remembered by end users, is that the anti-spam technologies should be not only designed and developed, but also deployed and used.

7 Acknowledgements

We would like to thank Prof. Fabio Massacci for many useful discussions and for suggesting the way to structure the comparison section.

References

[1] Banit Agrawal, Nitin Kumar, and Mart Molle. Controlling spam emails at the routers. In *Proceedings of the IEEE International Conference on Communications, ICC 2005*, volume 3, pages 1588–1592, 2005.

[2] Keno Albrecht, Nicolas Burri, and Roger Wattenhofer. Spamato – an extendable spam filter system. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[3] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '00*, pages 160–167, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-226-3. doi: http://doi.acm.org/10.1145/345508.345569.

[4] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. In G. Potamias, V. Moustakis, and M. van Someren, editors, *Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, ECML 2000*, pages 9–17, 2000.

[5] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine Spyropoulos, and Panagiotis Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based 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.

[6] Ion Androutsopoulos, Georgios Paliouras, and Eirinaios Michelakis. Learning to filter unsolicited commercial e-mail (Technical Report 2004/2). NCSR "Demokritos". Revised version. 2004.

[7] Ion Androutsopoulos, Evangelos Magirou, and Dimitrios Vassilakis. A game theoretic model of spam e-mailing. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[8] Hrishikesh Aradhye, Gregory Myers, and James Herson. Image analysis for efficient categorization of image-based spam e-mail. In *Proceedings of Eighth International Conference on Document Analysis and Recognition, ICDAR 2005*, volume 2, pages 914–918. IEEE Computer Society, 2005.

[9] Enrico Blanzieri and Anton Bryl. Evaluation of the highest probability svm nearest neighbor classifier with variable relative error cost. In *Proceedings of Fourth Conference on Email and Anti-Spam, CEAS'2007*, page 5 pp., 2007.

[10] P Boykin and Vwani Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4):61–68, 2005.

[11] A. Bratko, G. V. Cormack, B. Filipič, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 7(Dec):2673–2698, 2006.

[12] CAPTCHA. The CAPTCHA project. *http://www.captcha.net/ Accessed: 31.05.06*, 2005.

[13] Xavier Carreras and Lluís Márquez. Boosting trees for anti-spam email filtering. In *Proceedings of 4th International Conference on Recent Advances in Natural Language Processing, RANLP-01*, 2001.

[14] Jason Chan, Irena Koprinska, and Josiah Poon. Co-training on textual documents with a single natural feature set. In *Proceedings of the Ninth Australasian Document Computing Symosium (ADCS 2004)*, 2004.

[15] Paul Alexandru Chirita, rg Diederich Jö, and Wolfgang Nejdl. Mailrank: Using ranking for spam detection. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM 2005*, pages 373–380. ACM Press, 2005.

[16] Zhan Chuan, Lu Xianliang, Hou Mengshu, and Zhou Xu. A lvq-based neural network anti-spam email approach. *ACM SIGOPS Operating Systems Review*, 39(1): 34–39, 2005. ISSN 0163-5980. doi: http://doi.acm.org/10.1145/1044552.1044555.

[17] William Cohen. Learning rules that classify e-mail. In *Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access, MLIA '96*. AAAI Press, 1996.

[18] Gordon Cormack and Thomas Lynam. TREC 2005 spam track overview. *Available at plg.uwaterloo.ca/ gvcormac/trecspamtrack05/, Accessed: 31.05.06*, 2005.

[19] Gordon Cormack and Thomas Lynam. Spam corpus creation for TREC. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[20] Gordon V. Cormack and Andrej Bratko. Batch and online spam filter comparison. In *Proceedings of the Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[21] Wendy Cukier, Susan Cody, and Eva Nesselroth. Genres of spam: Expectations and deceptions. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences, HICSS '06*, volume 3, 2006.

[22] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. P2P-based collaborative spam detection and filtering. In *Proceedings of Fourth IEEE International Conference*

on Peer-to-Peer Computing, P2P'04, pages 176–183, 2004.

[23] Sarah Jane Delany, Padraig Cunningham, and Lorcan Coyle. An assessment of case-based reasoning for spam filtering. In *Proceedings of Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science (AICS '04)*, pages 9–18, 2004.

[24] Christine Drake, Jonathan Oliver, and Eugene Koontz. Anatomy of a phishing email. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[25] Mark Dredze, Reuven Gevaryahu, and Ari Elias-Bachrach. Learning fast classifiers for image spam. In *Proceedings of the Fourth Conference on Email and Anti-Spam, CEAS'2007*, 2007.

[26] Harris Drucker, Donghui Wu, and Vladimir Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054, 1999.

[27] Zhenhai Duan, Yingfei Dong, and Kartik Gopalan. Diffmail: A differentiated message delivery architecture to control spam. In *Proceedings of 11th International Conference on Parallel and Distributed Systems, ICPADS 2005*, volume 2, pages 255–259, 2005.

[28] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology - Crypto 92 Proceedings*, pages 139–147. Springer Verlag, 1992.

[29] Tom Fawcett. "in vivo" spam filtering: a challenge problem for data mining. *KDD Explorations*, 5(2):140–148, 2003. doi: http://doi.acm.org/10.1145/980972.980990.

[30] Gordon Fecyk. Designated mailers protocol. *http://www.pan-am.ca/dmp/draft-fecyk-dmp-01.txt, Accessed: 31.05.06*, 2003.

[31] FerrisResearch. The global economic impact of spam. report #409. *Available at http://www.ferris.com/get_content_file.php?id=364 Accessed: 13.06.06*, 2005.

[32] Giorgio Fumera, Ignazio Pillai, and Fabio Roli. Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research*, (7):2699–2720, 2006.

[33] Anurag Garg, Roberto Battiti, and Roberto Cascella. "May I borrow your filter?" exchanging filters to combat spam in a community. In *AINA 2006. 20th International Conference on Advanced Information Networking and Applications*, volume 2, 2006.

[34] Jennifer Golbeck and James Hendler. Reputation network analysis for email filtering. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[35] Luiz Henrique Gomes, Cristiano Cazita, Jussara M. Almeida, lio Almeida Virgí, and Jr. Wagner Meira. Characterizing a spam traffic. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 356–369, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-821-0. doi: http://doi.acm.org/10.1145/1028788.1028837.

[36] Joshua Goodman. IP addresses in email clients. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[37] Joshua Goodman and Robert Rounthwaite. Stopping outgoing spam. In *EC'04: Proceedings of the Fifth ACM Conference on Electronic Commerce*, 2004.

[38] Joshua Goodman and Wen-tau Yih. Online discriminative spam filter training. In *Proceedings of Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[39] Joshua Goodman, Gordon V. Cormack, and David Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):25–33, 2007.

[40] Paul Graham. A plan for spam. *Available at http://www.paulgraham.com/spam.html Accessed: 14.05.07*, 2002.

[41] Paul Graham. Better bayesian filtering. *Available at http://www.paulgraham.com/better.html Accessed: 12.07.06*, 2003. URL `http://www.paulgraham.com/better.html`.

[42] Galen A. Grimes. Compliance with CAN-SPAM act of 2003. *Communicationf of the ACM*, 50:55–62, 2007.

[43] Evan Harris. The next step in the spam control war: Greylisting. *Available at http://projects.puremagic.com/greylisting/ Accessed: 02.10.07*, 2003.

[44] Shlomo Hershkop. Behavior-based email analysis with application to spam detection. phD Thesis. *Available at www1.cs.columbia.edu/ sh553/publications/ Accessed: 12.07.06*, 2006.

[45] HoneyPot. Project honey pot: Distributed spam harvester tracking network. *Available at http://www.projecthoneypot.org/, Accessed: 07.06.06*, 2004.

[46] Geoff Hulten, Anthony Penta, Gopalakrishnan Seshadrinathan, and Manav Mishra. Trends in spam products and methods. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[47] ITU. ITU survey on anti-spam legislation worldwide. *Available at http://www.itu.int/osg/spu/spam/ Accessed: 31.05.06*, 2005.

[48] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[49] Jaeyeon Jung and Emil Sit. An empirical study of spam traffic and the use of dns black lists. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 370–375, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-821-0. doi: http://doi.acm.org/10.1145/1028788.1028838.

[50] Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[51] Benjamen Kuipers, Alex Liu, Aashin Gautam, and Mohamed Gouda. Zmail: zero-sum free market control of spam. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops, ICDCS 2005*, pages 20–26. IEEE Computer Society, 2005.

[52] Li Kun-Lun, Li Kai, Huang Hou-Kuan, and Tian Sheng-Feng. Active learning with simplified SVMs for spam categorization. *Machine Learning and Cybernetics*, 3:1198–1202, 2002.

[53] Chih-Chin Lai and Ming-Chi Tsai. An empirical performance comparison of machine learning methods for spam e-mail categorization. *Hybrid Intelligent Systems*, pages 44–48, 2004.

[54] Lorenzo Lazzari, Marco Mari, and Agostino Poggi. Cafe - collaborative agents for filtering e-mails. In *Proceedings of 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, WETICE'05*, pages 356–361, 2005.

[55] Honglak Lee and Andrew Ng. Spam deobfuscation using a hidden markov model. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005. URL `http://www.ceas.cc/papers-2005/166.pdf`.

[56] Barry Leiba, Joel Ossher, V. T. Rajan, Richard Segal, and Mark Wegman. SMTP path analysis. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005. URL `http://www.ceas.cc/papers-2005/176.pdf`.

[57] J. Levine and A. DeKok. Lightweight MTA authentication protocol (LMAP) discussion and comparison. *http://www.taugh.com/draft-irtf-asrg-lmap-discussion-01.txt, Accessed: 31.05.06*, 2004.

[58] Kang Li and Zhenyu Zhong. Fast statistical spam filter by approximate classifications. *SIGMETRICS Performance evaluation review*, 34 (1):347–358, 2006. ISSN 0163-5999.

[59] Kang Li, Calton Pu, and Mustaque Ahamad. Resisting spam delivery by tcp damping. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[60] Daniel Lowd and Christopher Meek. Good word attacks on statistical spam filters. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005. URL `http://www.ceas.cc/papers-2005/125.pdf`.

[61] Nicola Lugaresi. European union vs. spam: A legal response. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[62] Xiao Luo and Nur Zincir-Heywood. Comparison of a SOM based sequence analysis system and naive bayesian classifier for spam filtering. In *Proceedings of IEEE International Joint Conference on Neural Networks, IJCNN '05*, volume 4, pages 2571–2576, 2005.

[63] MAAWG. Messaging anti-abuse working group. Email metrics report. Third & fourth quarter 2006. *Available at http://www.maawg.org/about/ MAAWG-Metric_2006_3_4_report.pdf Accessed: 04.06.07*, 2006.

[64] Ben Medlock. An adaptive approach to spam filtering on a new corpus. In *Proceedings of the Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[65] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes? which naive bayes? In *Proceedings of Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[66] Eirinaios Michelakis, Ion Androutsopoulos, Georgios Paliouras, George Sakkis, and Panagiotis Stamatopoulos. Filtron: A learning-based anti-spam filter. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[67] Guoging Mo, Wei Zhao, Haixia Cao, and Jianshe Dong. Multi-agent interaction based collaborative p2p system for fighting spam. In *IAT'06. IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 428–431, 2006.

[68] Evangelos Moustakas, C. Ranganathan, and Penny Duquenoy. Combating spam through legislation: A comparative analysis of us and european approaches. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[69] Dhinaharan Nagamalai, Cynthia Dhinakaran, and Jae Kwang Lee. Multi layer approach to defend DDoS attacks caused by spam. In *MUE'07. International Conference on Multimedia and Ubiquitous Engineering*, pages 97–102, 2007.

[70] Cormac O'Brien and Carl Vogel. Spam filters: bayes vs. chi-squared; letters vs. words. In *Proceedings of the 1st international symposium on Information and communication technologies, ISICT '03*, pages 291–296, Dublin, Ireland, 2003. Trinity College Dublin.

[71] Patrick Pantel and Dekang Lin. Spamcop: A spam classification & organization program. In *Learning for Text Categorization: Papers from the 1998 Workshop*. AAAI Technical Report WS-98-05, 1998.

[72] So Young Park, Jeong Tae Kim, and Shin Gak Kang. Analysis of applicability of traditional spam regulations to voip spam. In *ICACT*

2006. The 8th International Conference on Advanced Communication Technology, volume 2, 2006.

[73] Matthew Prince, Benjamin Dahl, Lee Holloway, Arthur Keller, and Eric Langheinrich. Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. In *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[74] Calton Pu and Steve Webb. Observed trends in spam construction techniques: A case study of spam evolution. In *Proceedings of Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[75] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. In *SIGCOMM'06: Proceedings of the 2006 Conference on Aplications, Technologies, Architectures, and Protocols for Computer Communications*, 2006.

[76] Isidore Rigoutsos and Tien Huynh. Chung-kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (spam). In *Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004*, 2004.

[77] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*. AAAI Technical Report WS-98-05, 1998.

[78] Takamichi Saito. Anti-spam system: Another way of preventing spam. In *Proceedings of the 16th International Workshop on Database and Expert Systems Applications, DEXA 2005*, pages 57–61, 2005.

[79] Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine Spyropoulos, and Panagiotis Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In *Proceedings of Empirical Methods in Natural Language Processing, EMNLP-2001*, pages 44–50, 2001.

[80] Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine Spyropoulos, and Panagiotis Stamatopoulos. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6:49–73, 2003.

[81] Minoru Sasaki and Hiroyuki Shinnou. Spam detection using text clustering. In *Proceedings of International Conference on Cyberworlds, CW2005*, pages 316–319, 2005.

[82] Vincent Schiavone, David Brussin, James Koenig, Stephen Cobb, and Ray Everett-Church. Trusted e-mail open standard: A comprehencive policy and technology proposal for email reform. *http://www.cobb.com/spam/teos/, Accessed: 31.05.06*, 2003.

[83] D. Sculley and Gabriel M. Wachman. Relaxed online svms for spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 415–422, 2007.

[84] Larry Seltzer. Should senders pay for the mess we call e-mail? *eWeek, http://www.eweek.com/article2/0,4149,1273186,00.asp, Accessed: 31.05.06*, 2003.

[85] SenderID. Sender ID technology: Information for IT professionals. *Available at http://www.microsoft.com/mscorp/safety/technologies/senderid/technology.mspx, Accessed: 31.05.06*, 2004.

[86] Mikko Siponen and Carl Stucke. Effective anti-spam strategies in companies: An international study. In *Proceedings of HICSS '06*, volume 6, 2006.

[87] Nuanwan Soonthornphisaj, Kanokwan Chaikulseriwat, and Piyanan Tang-On.

Anti-spam filtering: a centroid-based classification approach. *Signal Processing*, 2: 1096–1099, 2002.

[88] SpamDefined. Spam defined. *http://www.monkeys.com/spam-defined/ Accessed: 31.05.06*, 2001.

[89] SPAMHAUS. The spam definition and legalization game. *Available at http://www.spamhaus.org/news.lasso?article=9, Accessed: 31.05.06*, 2003.

[90] SPAMHAUS. The definition of spam. *Available at http://www.spamhaus.org/definition.html, Accessed: 10.06.06*, 2005.

[91] Richard Daniel Twining, Matthew M. Williamson, Miranda Mowbray, and Maher Rahmouni. Email prioritization: reducing delays on legitimate mail caused by junk mail. Technical Report HPL-2004-5R1, HP Labs, 2004.

[92] SPF. FAQ. *http://openspf.org/faq.html Accessed: 31.05.06*.

[93] Xiao-Lin Wang and Ian Cloete. Learning to classify email: a survey. In *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*, pages 5716–5719, 2005.

[94] Zhe Wang, William Josephson, Qin Lv, Moses Charikar, and Kai Li. Filtering image spam with near-duplicate detection. In *Proceedings of the Fourth Conference on Email and Anti-Spam, CEAS'2007*, 2007.

[95] Gregory Wittel and Felix Wu. On attacking statistical spam filters. In *Proceedings of First Conference on Email and Anti-Spam, CEAS'2004*, 2004. URL `http://www.ceas.cc/papers-2004/170.pdf`.

[96] Matthew Woitaszek, Muhammad Shaaban, and Roy Czernikowski. Identifying junk electronic mail in microsoft outlook with a support vector machine. In *Proceedings of the 2003 Symposium on Applications and the Internet, SAINT 2003*, pages 166–169, 2003.

[97] Ching-Tung Wu, Kwang-Ting Cheng, Qiang Zhu, and Yi-Leh Wu. Using visual features for anti-spam filtering. In *Proceedings of IEEE International Conference on Image Processing, ICIP 2005*, volume 3, pages 509–512, 2005.

[98] Nariyoshi Yamai, Kiyohiko Okayama, Takuya Miyashita, Shin Maruyama, and Motonori Nakamura. A protection method against massive error mails caused by sender spoofed spam mails. In *Proceedings of the 2005 Symposium on Applications and the Internet, SAINT 2005*, pages 384–390, 2005.

[99] Chi-Yuan Yeh, Chih-Hung Wu, and Shing-Hwang Doong. Effective spam classification based on meta-heuristics. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, SMC 2005*, volume 4, pages 3872–3877, 2005.

[100] Wen-tau Yih, Joshua Goodman, and Geoff Hulten. Learning at low positive rates. In *Proceedings of the Third Conference on Email and Anti-Spam, CEAS'2006*, 2006.

[101] Le Zhang and Tianshun 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.

[102] Le Zhang, Jingbo Zhu, and Tianshun Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3 (4):243–269, 2004. ISSN 1530-0226. doi: http://doi.acm.org/10.1145/1039621.1039625.

[103] Wenqing Zhao and Zili Zhang. An email classification model based on rough set theory. In *Proceedings of the 2005 International Conference on Active Media Technology, AMT05*, pages 403–408, 2005.

[104] Feng Zhou, Li Zhuang, Ben Zhao, Ling Huang, Anthony Joseph, and John Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Proceedings of ACM/IFIP/USENIX International Middleware Conference, Middleware 2003*, 2003.

[105] Yan Zhou, Madhuri S. Mulekar, and Praveen Nerellapalli. Adaptive spam filtering using dynamic feature space. In *Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'05*, pages 302–309, 2005.

[106] Aaron Zinman and Judith Donath. Is Britney Spears spam? In *Proceedings of the Fourth Conference on Email and Anti-Spam, CEAS'2007*, 2007.

[107] Vasilios Zorkadis, M. Panayotou, and Dimitris A. Karras. Improved spam e-mail filtering based on committee machines and information theoretic feature extraction. In *Poceedings of IEEE International Joint Conference on Neural Networks, IJCNN '05*, volume 1, pages 179–184, 2005.