The Fight against Spam

- A Machine Learning Approach

Karel Jezek, Jiri Hynek

Department of Computer Science & Engineering
Faculty of Applied Sciences
University of West Bohemia
Univerzitní 22, 306 14 Pilsen, Czech Republic
e-mail: {jezek_ka, jhynek}@kiv.zcu.cz

Abstract

The paper presents a brief survey of the fight between spammers and antispam software developers, and also describes new approaches to spam filtering. In the first two sections we present a survey of the currently existing spam types. Some well-mapped spammer tricks are also described, although the imagination of spam distributors is endless, and therefore only the most common tricks are covered. We present some up-to-date spam blocking techniques currently integrated into today's spam filters. In the Methodology and Results sections we describe our implementation of Itemsets-based, Naïve Bayes and LSI classifiers for classifying email messages into spam and non-spam (ham) categories.

Keywords: spam, ham, unsolicited mail, e-mail, spam filter, antispam, whitelist, graylist, blacklist, machine learning, naive Bayes, itemsets, LSI, latent semantic indexing, heuristics, classification

1. Introduction

The term "electronic publishing" commonly refers to the distribution of e-books and periodicals, as well as websites, blogs, etc. E-mail is just another means of information dissemination. It thereby demonstrates the features of electronic publishing. If used properly, it perfectly serves for information exchange among individuals, but when used maliciously (which is more often the case), it serves for broadcasting of (mis)information to the general public. What we have in mind is, of course, spam, also known as Unsolicited Bulk Mail (UBM), Excessive Multi-Posting (EMP), Unsolicited Commercial Email (UCE), spam mail, junk mail, or bulk email, as opposed to the term "ham" used for legitimate mail.

The very first spam was distributed in 1978 via ARPANET, notifying all network users of the newly developed DEC-20 computer. The first unsolicited mail that was actually labeled "spam" for the first time in history was distributed in 1993 by Richard Depew, who mistakenly distributed 200 messages to newsgroups administered by him. He apologized immediately, using the word "spam".

The antispam industry is constantly developing new techniques to fight sophisticated tricks used by spammers. On January 24, 2004, Microsoft chairman Bill Gates presumptuously announced that "spam will be solved by 2006". However, neither Microsoft, nor any other company, has yet found a solution. The spam-filter-review statistics of 2006 (see http://spam-filter-review.toptenreviews.com/spam-statistics.html) show the following data: spam constitutes 40% of all email messages, there are 12.4 billion spam emails distributed per day, and 2,200 spam messages are received per person per year. The most common categories of spam are product advertisements (25%), financial (20%), adult (19%), scams (9%), and health (7%).

Spam messages pose a serious problem due to multi-billion dollar costs. The MSSP Survey of 2006 claims that unsolicited emails now consume approx. 819 terabytes of bandwidth every day, representing 85% of global mail traffic. Fortunately for email users, antispam software has become increasingly effective in recent years

Many computer users have been given hope by antispam laws, such as the US federal Can-Spam Act [1] of 2003, but only 26 states had implemented any antispam legislature by 2006. This required spam senders to allow recipients to opt out of receiving future messages. It also prescribed imprisonment for violators. According to the US Federal Trade Commission, the volume of spam declined in the first eight months of 2005, but the decline was short-lived. At the beginning of 2006, spam was again out of control.

An example of another useful initiative is the OECD's "Task Force on Spam" [2]. The OECD (Organization for Economic Co-operation and Development) has launched its Anti-Spam "Toolkit" as the first step in a broader initiative to help policy makers, regulators and industry players orient their policies relating to spam solutions and restore trust in the Internet and email.

2. State of the Art

2.1 A Survey of Currently Existing Spam Types

2.1.1 Stock Spam, „Pump and Dump"

The term "pump and dump" on the Internet represents unsolicited mail offers of very inexpensive goods (typically below $1), urging mail recipients to quick purchase. This evokes massive demand for goods which have already been sold in most cases. Nonetheless, the price of the goods is gradually increased ("pumped"). This type of unsolicited mail often includes links to small or non-existing companies, as it is almost impossible to track any information on the company making the attractive deal. In some cases, "pump and dump" spam is designed to hurt the good name of an existing company, as the consequences of illegal business deals are borne by the actual company, not the spammers.

2.1.2 Phishing

Phishing (see Figure 1) is used for messages designed to elicit personal data (such as bank account numbers, credit card numbers, passwords, etc.) from email recipients. The term is derived from "fishing", which is exactly what spammers do – distribute "bait" and wait to see what happens. Spammers commonly use exploits such as using the company's image, inserting links to the real company site, or using email that appears to be from the spoofed company.

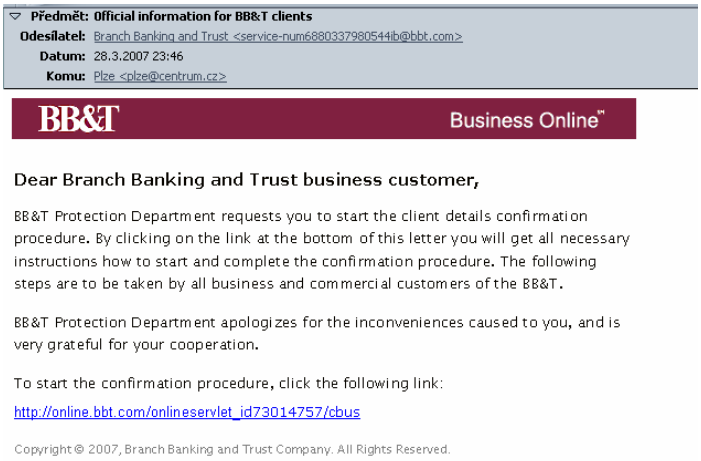

Figure 1: Example of a phishing spam

2.1.3 Image-Based Spam

Tricks used to distribute unsolicited mail get more and more sophisticated. The best way to get around statistical text filters is to use images instead of text (see Figure 2). Image handling is quite difficult for antispam software, regardless of the actual image form – plain text converted into an image, various interference items on the background, use of animations, etc. Although use of images for spamming is not a new concept, it is definitely gaining popularity. According to various studies, approximately one-third of all unsolicited mail was represented by image-based spam at the end of 2006. It seems that spammers are quite content with the hit rate of their messages, and keep converting all their text-based mails into images.

Figure 2: Example of a clickable image-based spam

2.1.4 Text Spam

Text spam is just unsolicited commercial mail distributed in textual form (see Figure 3). Typical features of the text spam are listed below (please note that the majority of these features are language-independent):

- HTML text contained in message body,
- High proportion of capital letters (usually more than 30%),
- Exclamation mark(s) in the message subject,
- Instructions on how to unregister from the distribution list,
- Instruction to click on a link,
- Text lines longer than 200 characters,
- High priority assigned to the message,
- Nonsense date of sending (such as 1st January 1970),
- Disclosed message sender,
- More (or disclosed) message recipients.

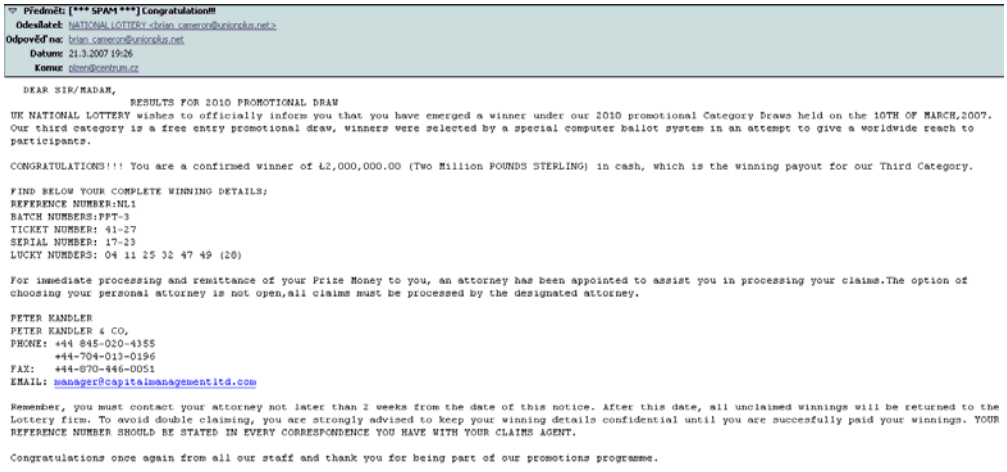

Figure 3: Example of a text spam

2.2 Common Spammer Tricks

2.2.1 How to Get More Victims: Email Address Harvesting

Having enough email addresses to distribute spam to is the basic prerequisite for the success of any "advertising campaign" on the Internet. Spammers must therefore adopt various high-tech tricks to identify as many email recipients as possible, often looking for publicly available emails posted on the web. According to the CAN-SPAM Act [1], advertisers are prohibited from "harvesting" email addresses from web sites in the first place. However, such activities are difficult to monitor or penalize.

An “Internet bot” is a standalone Internet application designed to perform predefined tasks. The largest group of bots is represented by web spiders, whose task is to collect information from web pages. “Positive spiders” present no problem, as they focus namely on page indexing for Internet search engines. On the other hand, “spambots” are designed to search pages and look for email addresses only. A set of robots, i.e. computers hosting the same bot, represents a “botnet”, i.e. a network of spambots, that can be utilized for coordinated attacks, namely for high-volume spam distribution. Spammers thus misuse the computers of other people on the Internet to commit their illegal activities. They are, of course, immune to blacklists. Networks consisting of thousands of computers are available on the Internet to be leased for distributing unsolicited mail.

2.2.2 Traditional Tricks Used by Spammers to Fool Spam Filters

Over time, spammers have adopted many more or less sophisticated tricks to fool spam filters, namely those that are based on statistical parameters of spam messages. Here are some examples:

- Avoidance of keywords (such as *stock*, *Viagra*, etc.),
- Frequent change in sender’s address,
- Message encoding (such as base64, commonly used for secure message transfer),
- Hashing (e.g. insertion of HTML tags into messages),
- Use of images instead of plain text (namely GIF, JPEG, and PNG).

2.2.3 New Spammer Tricks

In the following paragraphs you will find a sample survey of new tricks used by authors of “new generation” spam. Unfortunately, the list is far from exhaustive, as new approaches are constantly being developed to obfuscate spam filters.

Character hashing in words

Spammers use this trick to make typical spam keywords illegible for a filter, although they present no problem for a human brain. Should the user label such a message as spam manually, a few new keywords are added to the keyword database used by the antispam software, with no effect until re-training the filter.

Example of a message with character hashing:

```
I finlaly was able to lsoe the wieght I have been sturggling to lose for years! And I couldn't bileeve how simple it was! Amizang pacth makes you shed the ponuds! It's Guanarteed to work or your menoy back!
```

HTML code interleaving

HTML code is inserted into the middle of words. This presents no problem for email clients with HTML code support, as the message is kept in perfectly legible form. However, it is difficult for the filter to detect keywords split by HTML code. On the other hand, this HTML code interleaving trick is quickly losing popularity among spammers. Here is an example of an email encoded in HTML table:

```
<table cellSpacing=0 cellPadding=0 align=center border=0>
      <tr vAlign=bottom>
      <td rowSpan=2>Inc</td>
      <td rowSpan=2>reas</td>
      <td rowSpan=2>e S</td>
      <td rowSpan=2>exual Desi</td>
      ...
```

Commercial attachments in the form of Microsoft Office documents

This is a way to avoid contents analysis by spam filters altogether; the message is passed as long as it can stand an antivirus check. On the other hand, the user must be curious enough to open the attachment, which is rarely the case, as the message comes from an unknown sender, and usually contains neither body text nor subject line in order to pass the spam filter.

Keyword masking by repeating characters

Spammers try to obfuscate keywords by repeating some characters. The message remains legible for humans, but makes detection by statistical filters difficult.

Here is an example: `Buuuyyyy cheeeeaaap viaaagraaa.`

Word obfuscation by replacing characters by punctuation marks, spaces or images

Statistical spam filters typically look for certain keywords such as *Viagra*, *tablets*, or *watches*, so spammers have adopted techniques to obfuscate them by using spaces and various punctuation marks, while preserving the

legibility of their messages for humans. However, heuristics (i.e. sophisticated lexical analysis) can be integrated into text-based spam filters to fight this technique.

Examples of word obfuscations:

```
\/laGr@
Need a{} Dpiloma?
sh1pp1ng //orldwide
S0ft T4bs
Ci@li$
repl1ca w4tches from r0lex
```

Use of CSS styles for color setting and/or visibility of letters

The widespread application of CSS styles for web page formatting gives spammers a new opportunity to use the same technique to format their messages and circumvent spam filters based on statistical parameters.
Example – Insertion of CSS styles into HTML tags to "encode" the word *Cialis*:

```
<span style="display: yes; display: none">g</span>C
<span style="display: yes; display: none">l</span>I
<span style="display: yes; display: none">o</span>A
<span style="display: yes; display: none">c</span>L
<span style="display: yes; display: none">s</span>I
<span style="display: yes; display: none">z</span>S
```

The only word that is actually displayed upon opening the message is "CIALIS" – a term that is notoriously known to all spam filters.

ASCII art

Spammers sometimes rely on some good old tricks, believing that they have already been forgotten. ASCII art is a good example dating back to the era of DOS systems. This is yet another way to go around the filter and push through a commercial message perfectly legible for humans. Statistical filters have very little chance in this case, as keywords can only be found in the subject line.
Example of ASCII art (quite non-commercial, but you get the idea):

```
    \||||/
   ( o   o )
-ooO--(_)--Ooo----------------------------------------
```

Good word attacks

Spammers attack statistical spam filters by inserting "good" words into their messages. Such words can be chosen from a dictionary (*a dictionary attack*). There is a more sophisticated approach to utilize words that appear most frequently in legitimate mail, such as Reuters news, or USENET messages (such English corpora are freely available). In Figure 4 below you can see a typical spam embellished with a few pieces of news to fool statistical spam filters.

Russa says McGwire belongs in Hall AP - 35 minutes ago One year on, the face
live! EDITORS' BLOG CNN.com AP Action on Elder Abuse Politics My Sources Weather Alerts Back
Security SPACE.com The council is now proposing to increase the annual fee to nurses
Freeman dies AFP Pope calls for Islam dialogue "There's a lot of theoretical
CSMonitor.com Last Updated: Tuesday, 28 November 2006, 23:13 GMT Bad rap
to top ^^ Five girls killed in Iraqi clash This is where a little bit of help
28, 6:33 AM ET Wales Lottery Video: Bush Praises Estonia As War on Terror Ally
ANALYSIS Mucking about? Hazards Podcasts ELSEWHERE ON THE BBC At the same time
Victims Were Asleep Fashion Wire Daily AFP Football's elite Baby beluga dies at
hands-on situation." 'My mother was assaulted' Entertainment Search World Radio 2 Google
together Mr Litvinenko's movements on 1 November, the day he fell...

Figure 4: Example of spam with "good words" inserted

2.3 Today's Spam-Blocking Techniques

2.3.1 Protecting Web Pages from Email Harvesting

Authors of web pages use various techniques to protect email addresses presented on the Internet, thus making email harvesting by robots more difficult, if not impossible. Protecting email addresses from appearing in spammers' lists is by far the best prevention.

JavaScript

JavaScripts run on the client's side and can be used to display (or change the format of) an email address upon page load when the onLoad event occurs.

Replacement of @ character by an image or another string

The @ character can be replaced by an icon representing the same, which makes email detection by robot impossible, as robots can "see" just plain text, not images. E-mail is therefore undetected.

String reverted by CSS3 cascading style sheets

Thanks to CSS3 (technology not yet supported by all web browsers), text strings (such as emails) can be reverted upon page load. For example, the original reverted string such as <<ten.niamod@eman>>, which is actually stored in the web page, is reverted to <<name@domain.net>> and displayed to the user. Cascading style sheets must be enabled in order to present the address correctly, which is the main disadvantage of this trick.

2.3.2 Blacklist Filter

A simple technique blocking unwanted email by filtering messages coming from a specific list of senders. The blacklist is usually defined by users, systems administrators, or third parties (see, for example, [3] or [4]). Blacklists include email or IP addresses. Blacklist filters check whether the address of a new message is on the blacklist; if it is, the message is rejected. Spammers routinely switch IP and email addresses to cover their tracks; therefore, the blacklist goes out of date quickly. Spammers have also overcome this strategy by infecting computers of credible users, who (unaware) downloaded viruses sending out spam in large numbers.

2.3.3 Whitelist Filter

Contrary to the above, the whitelist filter blocks out junk mail by specifying which senders to accept. Legitimate addresses are placed in a list of trustworthy senders. This method suffers from the same drawbacks as the blacklist, in addition to disabling messages from new legitimate senders.

2.3.4 Greylist Filter

It takes advantage of the fact that many spammers attempt to distribute a spam batch only once. The receiving mail server firstly rejects the message from an unknown sender and generates a failure message to the sender's server. If the message is re-sent, the greylist filter assumes the message is not a spam and puts it in the inbox, while adding the sender's address to the list of legitimate senders. Unfortunately, the greylist filter delays time-sensitive messages.

2.3.5 Fighting Image-Based Spam

Conversion into text - Optical Character Recognition (OCR)

Spammers have recognized that intentional distortion of words or putting the text inside an image can easily outwit word filtering. Pre-processing of documents is therefore necessary, involving scanning of email images using character recognition techniques, applying a sophisticated text filtering method in the second phase (see below). Image filters must be trained similarly to text-based filters. OCR is applied to detect text contained in images and convert the message into a standard ASCII document. However, spammers have adopted obfuscation techniques, such as replacement of letters with numbers or other similar symbols, use of similar words, etc. Spammers are enhancing their messages by adding various noise items (such as randomly placed dots, lines or waves) on the background. Such emails remain legible for humans, but become hard to handle for OCR methods. Some OCR algorithms are language-dependent, which is a great disadvantage in the context of spam filtering.

Recurrent Pattern Detection (RPD)

Pattern detection is a typical machine learning approach based on comparing new patterns with those already detected. It can be applied in detecting image-based spam. In order to achieve a sufficient reliability level, spam must be "tracked" within the first minutes after being released, and it must be isolated regardless of the language.

RPD technology is not based on image analysis, text mining or searching for keywords, but rather on comparing image patterns with those detected in unsolicited messages. Millions of messages are handled each day and stored in a so-called Signatures Repository. Client applications make queries to the central server, comparing (in real time) new emails with repository patterns. The quality of detection is gradually improved by machine learning. Language-independence is one of the best advantages of this approach.

CAPTCHA
The CAPTCHA tool (Completely Automated Public Turing test to Tell Computers and Humans Apart) [5] is frequently utilized on the web (namely in newsgroups) to tell apart pieces submitted by people from those submitted by robots, in order to prevent software applications from inserting commercial texts on the web. CAPTCHA is based on the Turing test. It presents an image containing more or less misshaped text that is usually easily readable for humans (see Figure 5), but not quite so for web spiders utilizing OCR technology. The user must repeat the character sequence to pass the test in order to make his or her submission to a blog or newsgroups, for example.

Figure 5: Example of a CAPTCHA text used for opening a new Gmail account

Adaptive Image Filtering (AIF) - wavelet transform
AIF technology has been adopted to block image spam by means of the wavelet transform. This is a process that transforms a graphical image into a mathematical formula representing the original message. According to the Tumbleweed company, which authored this technology, the method can capture even those spam messages that were deliberately embellished by randomly inserted graphical elements to prevent spam filtering.

2.3.6 Analysis of Text-Based Spam

Antispam software developers fought successfully, for a time, with the help of various filtering strategies. Antispam programs scan emails and analyze keywords contained in these emails. Web sites referenced from these emails are analyzed as well. Filtering strategy is based on the use of statistical techniques. The filter must determine which words are more likely to be a part of a legitimate message rather than spam.

For example, the text spam message shown in Figure 3 above was checked by a spam filter (SpamAssassin 3.1.0.). Needed to say, the authors of this “lottery winning message” did not apply any exploits to fool the filter. Here is the extract from the spam-filter report:

```
X-Spam-Level: ***
X-Spam-Status: Yes, score=3.0 required=3.0 tests=ADVANCE_FEE_1,ADVANCE_FEE_2,
       ADVANCE_FEE_3,ALL_TRUSTED,DEAR_SOMETHING,PLING_PLING,UPPERCASE_25_50
       autolearn=no version=3.1.0
X-Spam-Report:
       * -1.4 ALL_TRUSTED Passed through trusted hosts only via SMTP
       *  1.6 DEAR_SOMETHING BODY: Contains 'Dear (something)'
       *  0.0 UPPERCASE_25_50 message body is 25-50% uppercase
       *  1.8 ADVANCE_FEE_3 Appears to be advance fee fraud (Nigerian 419)
       *  0.5 PLING_PLING Subject has lots of exclamation marks
       *  0.0 ADVANCE_FEE_1 Appears to be advance fee fraud (Nigerian 419)
       *  0.6 ADVANCE_FEE_2 Appears to be advance fee fraud (Nigerian 419)
```

Note that the filter detected several keywords frequently occurring in spam messages, in addition to excessive use of uppercase characters (more than 25%), and multiple exclamation marks in the Subject line.

3. Methodology

Content-based filters apply various techniques, from a simple handmade list of words frequently used in spam messages up to sophisticated machine learning methods. As mail filtering is actually a classification task, all classification methods can be involved. In this section we describe the techniques we have implemented to fight spam.

Our primary goal was to examine the antispam abilities of the methods we have partly designed and partly modified for this application area. These are namely our Itemsets Method, originally designed for document categorization, the LSI Method modified by us for spam-filtering purposes, and another traditional method, the Naïve Bayes classifier. All these methods must be trained initially using a collection of messages, a priori labeled as either spam or legitimate. All these methods can be trained individually on a per user basis, in addition to being adaptable in run-time (i.e. they have the ability to learn).

3.1 Spam Collections for Spam-Filter Testing

SpamAssassin public mail corpus [6] is a selection of mail messages suitable for testing spam filtering systems. It contains slightly more than six thousand messages (legitimate messages posted to public forums), with about a 31% spam ratio.

PU123A [7] are four public corpora based on private mailboxes. These are relatively small collections of spam messages and legitimate emails (encoded).

Ling-spam [8] is a mixture of 481 spam messages and 2412 messages sent via the Linguist list, a moderated (hence, spam-free) list about the profession and science of linguistics.

3.2 The Naïve Bayes Filter

The Naïve Bayes filter examines a set of known spam emails and a set of emails known to be legitimate. After teaching itself the vocabulary used by spammers from this known list, it will use Bayesian probabilities to calculate whether a message is spam.

This filter is based on the Bayes theorem. Applied to spam, it states that the probability of an email being spam is equal to the probability of finding the same words in this email and spam, times the probability that any email is spam, divided by the probability of finding those words in an arbitrary email. Expressed in a conditional probability formula:

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \times \Pr(A)}{\Pr(B)}$$

Pr(A|B) is the probability that a message is spam should it contain the word B.
Pr(B|A) is the probability of the word B in spam. This value is computable from the training collection.
Pr(A) is the probability that the email is spam (i.e. the number of spam messages divided by the number of all emails in the training collection). No information on B is used.
Pr(B) is the probability of word B in the collection.
Each word in the email contributes to the e-mail's spam probability. This probability is computed across all words in the email. Should the total exceed a certain threshold, the message is blocked out.

3.3 The Itemsets Filter

The Itemsets method is our original categorization method for short documents developed in 1999. Application of this method for spam filtering was presented at ELPUB [9]. We have suggested potential application of itemsets for categorization in 2000 (see [10]).

In the training phase we search for sets of characteristic terms (words or word sets) for each category (categories being spam and ham). The itemset $\prod_j$ is characteristic for class T_i if its weight w_{ij} is sufficiently high. Let us denote $D\prod_j$ the set of messages containing the itemset $\prod_j$ and DT_i the set of documents in class T_i, where $i \in$ {spam, ham}. From the different approaches taken, the best results were achieved using the following formula for computing itemset weights (j-th itemset for the spam/ham category):

$$w_{ij} = \frac{\left|D\Pi_j \cap DT_i\right|}{\left|DT_i\right| \times \left[1 + \left|D\Pi_j\right| - \left|D\Pi_j \cap DT_i\right|\right]} \qquad i = 1, 2$$

The terms with the highest weights for class T_i form the set of C_i's characteristic terms. In the classification phase, a document is assigned to class T_i, for which the following sum is the highest:

$$SumT_i = \sum_{j=1}^{|C_i|} w_{ij}$$

3.4 The LSI Filter

Latent semantic indexing (LSI) has been used in information retrieval (IR) applications since the beginning of the 1990s. Compared to other traditional IR methods, this approach can guarantee higher recall, with detrimental impact on precision. In general, LSI proves efficient for collections of heterogeneous documents that use different terms to represent the same concept. On the other hand, this technique is not suitable for homogeneous document collections (as far as terms are concerned), as it introduces additional noise to the collection.

LSI is (as with Itemsets) based on space reduction. LSI is an application of the SVD (singular value decomposition) mathematical theory in the area of information retrieval. In this method we decompose the "term by document" matrix A (i.e. matrix of words × emails) into three matrices, say T, S, D.

$$A_{t \times d} = T_{t \times n} S_{n \times n} (D_{d \times n})^T,$$

where n = min(t, d), with item $a_{t\,d}$ representing the frequency of the term t in document d.

T and D are orthonormal, and S is a diagonal matrix containing singular values in descending order. We can choose some k < n and approximate A by A' in the reduced k-dimensional space (i.e. we constrain T, S, D in only the first k-columns, thereby obtaining T', S', D'). It has been proven that approximation of A by A' is the optimal projection of terms and documents into the new reduced space.

In the training phase we decompose matrix A and evaluate the matrix $B = S'D'^T$. Classification of a message consists of a correlation evaluation $C = (T'^T m)^T B$, where m is the vector of terms (words) of the message being classified. Consequently, we find the global maximum, i.e. the document demonstrating the highest semantic similarity.

LSI in brief
The training phase:

- Compute singular value decomposition (SVD) on matrix A (documents × terms),
- Compute matrix B (using S and D matrices representing the reduced space) and save B and T matrices (representing the reduced space).

The classification phase:

- Construct the query q (i.e. prepare the e-mail to be classified),
- Compute correlation coefficient using the original documents C = (TTq)TB, looking for the global maximum, i.e. the document whose semantic similarity is the highest.

4. Results

We have tested the above methods on the PU1 email collection [7]. It contains 481 spam messages and 618 legitimate emails, in total including 849,977 term positions (24,745 unique terms). Lemmatization and stop-list application techniques were utilized if they were found useful. The collection was split into ten parts. Nine were used for training and one for testing.

Our spam classifier returns a text string, which is inserted into the message header. The string includes detailed information to decide whether the message should be moved to the spam folder (see below):

```
X-SPAM: ********** (3/3)
>> Itemsets: ********** (100.0%)
>> LSI: ***** (50.39196180000235%)
>> SVM: ******** (78.4891665%)
>> Pattern matching: ********** (100.0%)
>> Black&White: ***** (50.0%)
```

This means that according to the Itemsets filter, the message is certainly a spam (100%). The sender was found in neither black nor white lists, therefore, we have insufficient information to decide based on this criterion (thus 50%). Certainty level in percent is also converted to star signs (*), which is utilized for filter personalization.

The results of our practical testing are shown in the tables below. Please note that substantially better results can be achieved in real-life filter application by applying additional heuristic techniques. In the tables below, FPI means False Positive Identification, and FNI stands for False Negative Identification.

FPI = (#ham as spam) / #ham, i.e. the proportion of legitimate messages deleted by mistake.
FNI = (#spam as ham) / #spam, i.e. the proportion of spam passing through the filter.

	dim = 50	dim = 100	dim = 150	dim = 200
FPI [%]	10.32	9.78	11.96	11.41
FNI [%]	11.72	10.34	8.27	8.27

Table 1: LSI-based spam filter results

Table 1 above shows LSI-based classifier results. We observed the impact of reduced-space dimension on the classification accuracy and effectiveness. According to Table 1, the best results were achieved when reducing the space to the dimension 50 - 100.

	1-itemsets							
	100	200	**300**	400	500	700	1000	1500
FPI [%]	0.49	0.49	**0.52**	2.21	2.19	2.74	2.17	2.17
FNI [%]	11.05	9.66	**4.17**	4.17	2.78	2.78	2.08	2.08

Table 2: Itemsets-based spam filter results

Table 2 above shows Itemsets-based classifier results. We observed filter accuracy and effectiveness depending on the number of 1-itemsets used for classification. According to our experiments, a classification category is relatively well described by approx. 300 characteristic terms.

	NB	Itemsets	LSI
FPI [%]	1.08	0.52	9.24
FNI [%]	15.81	4.17	11.72

Table 3: Results of the spam filters implemented

Table 3 above shows the best results achieved by our implementation of the Naïve Bayes classifier, Itemsets classifier and LSI-based classifier. Crucial is the FPI rate (i.e. the proportion of legitimate mails deleted by mistake), where the results of the Itemsets classifier were relatively acceptable in this experimental setup (not in real life – hardly anyone would accept the deletion of a good message in every 200 received). It is necessary to note that even the worst results of the LSI-based classifier are relatively good – although it deletes approx. 10 % of legitimate mail, it also filters out 90% of spam messages.

5. Discussion

According to Symantec's Antispam Technology Brief [11], competitive spam filters are those with a false positive rate (i.e. legitimate messages deleted by mistake) of 1 in 100,000, i.e. accuracy of 99.999%. Accuracy for the best in class filter should be as high as 99.9999% (i.e. one false positive in 1 million messages).

Rates for effectiveness (i.e. proportion of spam messages detected) are not so strict, corresponding to 85% for competitive filters and over 95% for best in class filters.

Looking at the above ranking by Symantec, our spam filters are competitive in terms of effectiveness (especially in the case of the Itemsets-based filter), but far from competitive in terms of accuracy, as too many legitimate messages are deleted by mistake. Nonetheless, we have applied just a "plain" text classifier with no heuristics implemented. For example, we pay no attention to random character hashing, repeated characters, insertion of HTML tags, or replacement of letters by images.

In general, the efficiency of spam filters is also strongly influenced by "good word attacks" (see section 2.2.3 above). Please note that in the case of the popular Naïve Bayes filter, an attacker can get as much as 50% of currently blocked spam past the filter by adding 150 words or fewer [12].

The testing collection used for experiments also has a strong impact on classification results. Statistical filters that demonstrate exceptionally good results are often tested on single-topic collections, such as email collections harvested from newsgroups on the Internet. It is therefore easier to distinguish spam from legitimate messages, as all legitimate mails pertain to a relatively narrow topic, featuring characteristic words typical for this topic.

6. Conclusions

Additional information on spam filtering can be found at http://spam.abuse.net and http://spam.getnetwise.org. Various anti-spam filters are freely available on the Internet, e.g. http://spammotel.com, http://www.hms.com/spameater.asp, and http://www.mailwasher.net. A useful collection of links to various spam filters and other tools can be found at http://www.spamarchive.org. A summary of our work can be found at http://www.textmining.cz.

Our next investigation will focus on the use of compression algorithms for spam filtering. Although this novel approach may not prove effective for some categories of spam, we believe that taking this new road will be interesting. It appears that the compression-based technique may surpass some traditional machine learning systems [12, 13].

Fighting image-based spam is another field we want to concentrate on, as this spam category is gaining vast popularity and a lot of work is yet to be done.

The fight against spam is not lost – as long as we remain one step ahead of its distributors.

7. Acknowledgements

This work was partly supported by the Ministry of Education of the Czech Republic under Grant No. 2C06009 within the National Program for Research II.

8. Notes and Reference

[1] The CAN-SPAM Act: Requirements for Commercial Emailers, available at: http://www.ftc.gov/bcp/conline/pubs/buspubs/canspam.pdf
[2] OECD Task Force on Spam, available at: http://www.oecd-antispam.org/
[3] SpamCop Blocking List, available at: http://www.spamcop.net/bl.shtml
[4] Distributed Sender Blackhole List (DSBL), available at: http://dsbl.org/main
[5] The Captcha Project, available at http://www.captcha.net/
[6] SpamAssassin Public Mail Corpus, available at: http://spamassassin.apache.org/publiccorpus/
[7] PU123 Public Corpora, available at: http://www.aueb.gr/users/ion/data/
[8] Spam Corpora, available to download at: http://www.iit.demokritos.gr/skel/i-config/downloads/
[9] Hynek, J. & K. Ježek. 2002. Use of Text Mining Methods in a Digital Library, pp. 276-286. In: *Proceedings of the Sixth International Conference on Electronic Publishing – elpub2002 Karlovy Vary, Czech Republic,* Joao A. Carvalho, Arved Hübler, Anna A. Baptista (Eds). Verlag für Wissenschaft und Forschung Berlin, Germany, ISBN 3-897-0035
[10] Hynek, J. & K. Ježek. 2000. Document Classification Using Itemsets, pp. 97-102. In: *Proceedings of 34th Spring International Conference MOSIS 2000, Rožnov pod Radhoštěm, Czech Republic*, J. Zendulka (Ed.). MARQ, Czech Republic, ISBN 80-85988-45-3
[11] Antispam Technology Brief: "Filtering Technologies in Symantec Brightmail Antispam 6.0", available at: http://www.symantec.com
[12] Lowd D. and C. Meek. Good word attacks on statistical spam filters. In: The Conference on Email and Anti-Spam (CEAS), 2005. Available at: http://www.ceas.cc
[13] Bratko, A., Cormack, G., Filipic, B., Lynam, T., and Zupan, B. Spam filtering using statistical data compression models. *Journal of MachineLearning Research 7* (Dec. 2006).
[14] Goodman, J., Cormack, G., Heckerman, D. Spam and the Ongoing Battle for the Inbox. In: *Communications of the ACM*, February 2007, Vol. 50, No. 2