An Intelligent Automatic Hoax Detection System

Conference Paper · September 2009

DOI: 10.1007/978-3-642-04595-0_39 · Source: DBLP

3 authors:

Marin Vuković
University of Zagreb
38 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Krešimir Pripužić
University of Zagreb
27 PUBLICATIONS **299** CITATIONS
SEE PROFILE

Hrvoje Belani
University of Zagreb
32 PUBLICATIONS **89** CITATIONS
SEE PROFILE

An Intelligent Automatic Hoax Detection System*

Marin Vuković[1], Krešimir Pripužić[1], Hrvoje Belani[1]

[1] University of Zagreb, Faculty of Electrical Engineering and Computing,
Department of Telecommunications, Unska 3, HR-10000 Zagreb, Croatia
{marin.vukovic, kresimir.pripuzic, hrvoje.belani}@fer.hr

Abstract. Although they sometimes seem harmless, hoaxes represent not-negligible threat to individuals' awareness of real-life situations by deceiving them, and at the same time doing harm to the image of their organizations, which can lead to substantial financial losses. Spreading of hoaxes also influences the normal operating regime of networks and the efficiency of workers. In this paper we present an intelligent automatic hoax detection system based on neural networks and advanced text processing. In the developing of our system we use a database with real-life e-mail hoaxes, and an additional database with real-life e-mail messages. At the end we give brief experimental evaluation of the hoax detection system and comment the results.

Keywords: Hoax detection, e-mail classification, n-grams, self-organizing map, feed forward neural network, experimental evaluation.

1 Introduction

Hoaxes (the term origins from: hocus to trick) are more or less present throughout the entire history of mankind. Usual intention of hoax creator is to persuade or manipulate other people to do or prevent pre-established actions, mostly by using a threat or deception [1]. These intentions usually rely on empathy and abuse the human need for helping other people. Hoax creators want that their messages be read and forwarded to the largest possible number of victims. In today's world, hoaxing seems to find a fruitful ground in the new and emerging information and communication technologies (ICT), like e-mail, instant messaging, internet chats and mobile messaging.

Although hoaxes are not created to make technical damages to computer programs and operating systems, they can lead victims to damage their programs or systems, destroy reputation of them and their companies, coworkers and friends, or even to produce some financial losses. According to [2], which is considered to be the world's most widely quoted research on computer crime for years, various financial frauds result in an average reported loss of close to $500,000 per company. The share of

*This work was carried out within the research project "Content Delivery and Mobility of Users and Services in New Generation Networks", supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

hoaxes in these frauds is not pointed out specifically, but the hoax-related losses, like insider abuse of networks and phishing, are mentioned and briefly analyzed in the study. The spreading of hoaxes also influences the normal operating regime of network and efficiency of workers. Therefore, taking care of hoaxes is very important in order to minimize their impact on the different security-related frauds and misuses.

In science of memetics, a controversial research field that transcends psychology, biology, anthropology, and cognitive science, the term of hoaxes is sometimes referred to as a "virus of the mind" [3], mainly because of its ability to self-replicate, adapt, mutate, and persist in the human mind. They stay in the human mind and provoke the "infected" individuals to pass them to other individuals.

The next section surveys related work in the fields of text classification and hoax detection. The third section proposes a new hoax detection system based on artificial neural networks, while the fourth section describes a pre-processing of text, which is a very important part of the whole system. The fifth section presents and comments results of an experimental evaluation of implemented automatic hoax detection system. The sixth section gives conclusion and future work.

2 Related Work

Although the main definition of hoaxes origins from e-mail communication [4], in general it is every electronic message that contains bogus information with malicious intention to mislead its receiver. Such bogus information can be represented as textual, graphical, audio, and/or other multimedia content, e.g. fake virus warnings and various photo manipulations. In this work we focus on automatic detection of textual hoaxes, like e-mails [5], short message service (SMS) messages [6], and messages in Internet chats and forums.

When talking about unwanted e-mail messages, it is necessary to distinguish between hoaxes and unsolicited commercial e-mails, known as spam. Spam is, in its nature, created in order to sell a certain product or service, and therefore the usual expectation is that spam offers information which is exaggerate and not entirely true. On the other hand, hoax has a purpose of deceiving an average user and making him to believe in fake information it provides. A significant research work done in [7] addresses technological, organizational, behavioral, and legislative anti-spam measures, which unfortunately cannot be automatically applied to hoaxes.

The research area of automatic hoax detection gained a significant interest in the last decade, but with the partial results and solutions that are based on different approaches, e.g. heuristics, traffic analysis, etc. [1]. Authors of [4] developed a service that receives and evaluates e-mail messages that users forward when they suspect a hoax. Their approach and results are interesting, but not applicable for real-time hoax detection, in which we are interested. Furthermore, it is more difficult to detect if an unsuspected message is hoax.

For the classification of text, mostly unsupervised methods such as self-organizing maps (SOM) are used [8] [9], but there are also examples of using supervised learning methods [10] [11] [12]. The SOM architecture is very appropriate for this purpose, because it is able to classify the text according to the similarity of input patterns

which represent the e-mail text. However, in order to avoid poor classification results, the SOM input patterns, which represent the text, must be coded in a proper manner. Since we are interested in distinguishing hoaxes from regular e-mails, using only SOM, as done in [8] and [9], is not appropriate because it would classify a text into several clusters that contain both hoaxes and regular e-mails, depending on the input pattern coding scheme. Authors of [12] used only supervised learning for classification of e-mail messages to several groups. Since we are trying to distinguish hoaxes from regular e-mails only, our system can be more precise in performing its task. Besides the detection of hoax messages, we further improve our system to classify detected hoaxes as well. This is appropriate for various applications, such as improving the system in means of automatic learning and dealing with false positives which is discussed in the further text. Therefore, our solution combines both supervised and unsupervised learning. The supervised learning is used for distinguishing hoaxes from regular e-mails while unsupervised learning is used for hoax message classification.

3 Hoax Detection System

Our hoax detection system is composed of several modules as shown in Figure 1. After the preprocessing of e-mail content, which is described in the next section, the vector containing numeric representation of a single e-mail is presented to the hoax detector module.

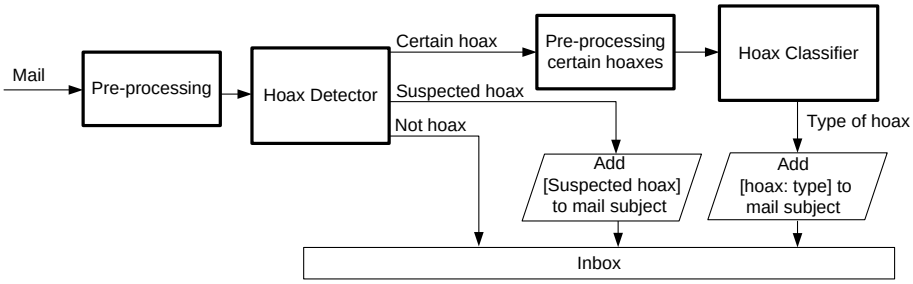

Figure 1 – Hoax Detection System

The hoax detector module is realized by a feed-forward artificial neural network whose task is to distinguish regular e-mails from e-mails containing hoaxes (or hoax e-mail in short). In order to accomplish this task, the network had to be learned with both regular and hoax e-mails using supervised learning technique. To train and test the neural networks we have used a collection of 298 hoax and 1370 regular e-mail messages. These messages were in the plain text form. The size of the hoax collection was 382 kilobytes, while the size of our mail collection was 2.92 megabytes. Learning set was composed as follows:

$$Input: \left(ntf_{1,d} \cdot idfb_{1,d}\right), \left(ntf_{2,d} \cdot idfb_{2,d}\right) \ \ldots \ \left(ntf_{1369,d} \cdot idfb_{1369,d}\right),$$

$$Output: \ \ (hoax, \neg hoax\ ),$$

where $idfb_{i,d}$ denotes the inverse document frequency in the both collections.

The learning was performed using backpropagation algorithm on the network consisting of the following three layers:

- Input layer with 1369 input neurons,
- Hidden layer with 100 neurons, and
- Output layer consisting of two neurons.

The input layer size corresponds to the total number of important *n*-grams, which are obtained in the pre-processing phase as described in the next section. The purpose of the two output neurons is to see whether an incoming e-mail is certainly hoax, is suspected to be hoax, or is certainly not hoax. This functionality is achieved by activating only one neuron in the case when network concludes that an e-mail is certainly hoax, and activating other neuron otherwise. However, if the network is unsure, both neurons will be activated with low certainty and further checking is required. In this context, we define low certainty as activation of less than 0.75. Furthermore, if only one neuron is activated with certainty lower than 0.5 we also conclude that further checking is required, in order to prevent false positive and/or false negative results.

The learning set consisted of 1668 patterns, of which 298 were known hoax e-mails, and others were regular e-mail messages. Once the learning is finished, the network is able to decide whether a pattern at the input is certainly hoax, is suspected to be hoax, or is certainly not hoax.

As we can see in Figure 1, if an incoming e-mail at the input of hoax detector module is recognized as certainly not hoax, it will be forwarded to the inbox. If an e-mail is suspected to be hoax, the phrase „suspected hoax" will be concatenated to its subject. In this way, the reader will be informed about the suspicion and will be able to proceed to the mail with caution. Additionally, if enabled, our system offers an option that users read and categorize e-mails as hoaxes or not. This way, any user can enhance the system performance by expanding the known hoax database. Finally, if the hoax detector is sure that the e-mail is hoax, the system is able to classify it in a group with similar hoaxes. This approach has the two main advantages:

1. The system continues to build the hoax database which is very useful for further system enhancements,
2. Users can see that a hoax belongs to a certain group of wide-spread, known hoaxes, thus improving general consciousness about hoaxes and risks regarding them.

The main task of hoax classifier is to classify hoax e-mails. In order to do so, all such e-mails must be pre-processed as described in the next section, in such a way that the input vector contains only *n*-grams from collection of hoax e-mails, and not from the collection of regular e-mail messages. A self organizing map is used as the hoax classifier. As this type of network architecture requires unsupervised learning, the patterns do not have to include an output of the SOM. The patterns were formed as follows:

$$(ntf_{1,d} \cdot idfh_{1,d}), (ntf_{2,d} \cdot idfh_{2,d}) \ \ldots \ (ntf_{1880,d} \cdot idfh_{1880,d}),$$

where $idfh_{i,d}$ denotes the inverse document frequency in the hoax collection.

The SOM used for classification has 1880 input layer neurons and 20 neurons in the output layer. The training is done with Kohonen algorithm with the training set consisting of 298 known hoax messages. The network results are interpreted by winner neuron (*Winner Takes All*). Our experimental results prove this to be suitable for the purpose, because the goal of the SOM is to classify a hoax in the group with the most similar hoaxes.

4 Text Pre-Processing

As we explained in the previous section, we use two different neural networks in the system: a self-organizing map (SOM) and a feed-forward neural network. We use text processing methods to reduce the number of input neurons in both of these networks. This is a very important step for the whole system because it directly improves its performance by significantly reducing the number of input neurons in the neural networks. In this section we explain the text processing approach we use as a pre-processing step in our system.

Many of the regular e-mail and hoax messages in our collections were written in two or even more languages, but mostly in English or/and Croatian. Because of the mixture of different languages in a single document, we could not use stemming and lemmatization which are the standard text processing methods. In short, stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving reduction of a word to a common base form, while lemmatization achieves the same goal with the use of vocabulary [13] and morphological analysis of words [14]. Instead of these methods, we use *n*-gram (i.e. *k*-gram) tokenization [14], which is a process of breaking text to n-grams, where an *n*-gram is a sequence of *n* characters. For example, sentence *"Crni cvorak skakuce na sirokoj grani"* breaks into the following *3*-grams: {*crn, rni, cvo, vor, ora, rak, ska, kak, aku, kuc, uce, sir, iro, rok, oko, koj, gra, ran, ani*}.

Actually, we have to normalize text before *n*-gram tokenization. This is done by removing capitalization, punctuation and diacritics. Removing of capitalization is the first necessary step in our text normalization. This way we ignore the difference among same words that are capitalized differently. For example, we treat words "*Crni*" and "*crni*" equally. The second step in the normalization is to remove special characters and punctuation from text. It is very common to write text in Croatian without using the diacritics. For example, writing *"Crni cvorak skakuce na sirokoj grani."* instead of the proper sentence *„Crni čvorak skakuće na širokoj grani."* Therefore, the removing of diacritics is the third necessary step in the normalization. We summarize our text processing approach as follows:

1. Text Normalization
    a. Removing of capitalization
    b. Removing of special characters and punctuation
    c. Removing of diacritics
2. Text Tokenization
    a. *n*-gram tokenization

In the process of tokenization, we limit the size of *n*-grams to $n = 3 \ldots 8$. The shorter *n*-grams are too general and we have excluded them because their entropy is very low. On the other side, the longer *n*-grams are too specific for belonging document, and we have excluded them because they are not appropriate for the generalization among different documents. The total number of such *n*-grams in the both collections was 79448.

This is quite a large number of *n*-grams to directly use them as inputs of our neural networks. That is why we need to reduce their number such that we include only those *n*-grams that are actually important for pre-processing of text. For this purpose we use document frequency df_t, defined to be the number of documents in a collection that contains term *t*. As we can see in Figure 1, we use two different pre-processing modules: mail pre-processing and pre-processing of certain hoaxes. The only difference among these two modules is in a different selection of the **important *n*-grams**.

In the mail pre-processing, we discard *n*-grams that are rare in the hoax collection $df_t(hoax) < 15$ or very common in the both collections $df_t(hoax+mail) > 50$. The number of important *n*-grams (and input nodes in the first neural networks) obtained in this way is 1369. We drop common *n*-grams (stop words) because they cannot help us in distinction among different documents, while rare *n*-grams (discriminative words) cannot help the neural network in generalization. Because of the same reason, in the pre-processing of certain hoaxes, we discard *n*-grams that are rare $df_t(hoax) < 10$ or very common $df_t(hoax) > 20$ in the hoax collection. This time the number of obtained important *n*-grams is 1880. It is important to notice that the thresholds are lower in the second pre-processor. The main reason for this is the second neural network, which is a classifier, and therefore we need more rare *n*-grams as inputs to distinguish between different hoaxes.

At this point, we will explain how we create input values for the neural networks in the system. For an incoming e-mail, we apply the first two steps of our text processing approach (i.e. the normalization and tokenization). Then for each important *n*-gram in the e-mail we calculate product of its *normalized term frequency* and *inverse document frequency* values:

$$ntf_{t,d} \cdot idf_{t,d} = \left(a + (1-a)\frac{tf_{t,d}}{mtf_d}\right) \cdot log\frac{N}{df_t},$$

where $a = 0.4$ is the standard value of smoothing term, N is a number of documents in the related collection, $tf_{t,d}$ is *n*-gram frequency in the e-mail, and $mtf_d = \max_{\tau \in d} \mathrm{tf}_{\tau,d}$ is maximal *n*-gram frequency of all *n*-grams in the e-mail. These $ntf_{t,d} \cdot idf_{t,d}$ values are inputs for the neural networks.

5 Experimental Evaluation

First we evaluate the performance of the hoax detector module. After training the module is capable to distinguish all hoaxes from regular e-mails, which were contained in the training set, as expected. However, if a new e-mail is received at an input, it is classified according to its similarity with the "known" e-mails. Thus, the

results are shown in the Table 1. Obviously, the main issue concerns suspected hoaxes, i.e. e-mails which may or may not be hoaxes. This is because they contain words which are common in both hoaxes and regular e-mails.

False positives	Suspected hoax	False negatives	Correct
4,90%	19,70%	1,54%	73,86%

Table 1 – Performance results of the hoax detector module

The evaluation results of the hoax classifier module are the following. After training the SOM, the known hoax messages are divided in 20 groups, analogue to number of output neurons. The results for the four most common hoax groups are presented in Table 2. The outcome of the classifier could be altered by modifying the number of SOM output neurons if necessary. However, this has proven to be suitable for our purpose.

Group ID	Hoax theme	number of messages	percentage in hoax dataset
Group 6	Chained letters – prayers (in Croatian)	26	8,70%
Group 9	Chained letters – prayers (in English)	21	7%
Group 11	Asking help for surgery (in Croatian)	20	6,70%
Group 17	Warning recipients about something (in Croatian)	20	6,70%

Table 2 – The four most common hoax groups

6 Conclusion and Future Work

This paper proposes intelligent hoax detection system based on artificial neural networks for which the data have to be pre-processed using information retrieval methods. In experimental evaluation, we showed that it is possible to detect and classify hoax messages successfully, at least to some extent. The proposed system has the ability to distinguish and classify hoax messages by comparing them against known hoax messages which usually contain similar patterns. However, if a new hoax message would appear, which does not have any similarity with the ones contained in the training set, the proposed system would not be able to detect it. The main issue with hoax messages in general is that they could be very similar to regular e-mail messages and it is difficult to distinguish whether their content is true or not, even to a human. As future work, we plan to develop a technique which could further evaluate the message content thus lowering the number of false positives and, especially, suspected hoax messages.

Furthermore, implemented system can also be applied to automatic hoax detection in SMS messages, which becomes one of raising issues in the evolving field of value added services (VAS) in telecommunications. Nevertheless, this kind of system evaluation would require an easy-manageable dataset of SMS hoaxes, as well as regular SMS messages, which is not trivial to acquire.

References

1. Hernandez, J.C., Hernandez, C.J., Sierra, J.M., Ribagorda, A.: A First Step towards Automatic Hoax Detection. In: Proceedings of the International 36th Annual Carnahan Conference on Security Technology, IEEE, Piscataway NJ, pp. 102--114 (2002)
2. Richardson, R.: 2008 CSI Computer Crime & Security Survey. Computer Security Institute, San Francisco, CA, URL: http://www.gocsi.com/ (2008)
3. Brodie, R.: Virus of the Mind: The New Science of the Meme. Integral Press, USA, 1st edition (1995)
4. Petković, T., Kostanjčar, Z., Pale, P.: E-Mail System for Automatic Hoax-Recognition. In: XXVII. International Convention MIPRO 2005 Bd. CTS & CIS, ISBN 953–233–012–7, Opatija, Croatia, pp. 117--121 (2005)
5. Sakkis, G.: Learning How to Tell Ham from Spam. In: Crossroads, Volume 11, Issue 2, ISSN:1528-4972, ACM, New York, NY, USA (2004)
6. „SMS Hoax Causes Traffic Congestion“. textually.org: all about texting, SMS and MMS, URL: http://www.textually.org/textually/archives/2005/08/009494.htm. Accessed on: March 2009
7. Schryen, G.: Anti-Spam Measures - Analysis and Design. ISBN: 978-3-540-71748-5, Springer-Verlag, Berlin/Heidelberg (2007)
8. Kim, H-D., Cho, S-B.: Application of Self-Organizing Maps to Classification and Browsing of FAQ E-mails. PRICAI 2000 Workshop Reader, LNCS 2112, Springer-Verlag Berlin/Heidelberg, pp. 44--55 (2001)
9. Merkl, D., Rauber, A.: Document Classification with Unsupervised Artificial Neural Networks. In: F. Crestani, & G. Pasi (Eds.), Soft computing in information retrieval, Physica-Verlag, Wurzburg (Wien), pp. 102--121 (2000)
10. Jevtić, D., Car, Ž., Vuković, M.: Location Name Extraction for User Created Digital Content Services. In: Proceedings of the 11th International Conference Knowledge-Based Intelligent Information and Engineering Systems (KES 2007), Part I, XVII Italian Workshop on Neural Networks (WIRN 2007), LNCS/LNAI 4692 (0302-9743), Springer-Verlag, Berlin/Heidelberg, pp. 623--630 (2007)
11. Bin, Cui, B., Mondal, A., Shen, J., Cong G., Tan, K-L.: On Effective E-mail Classification via Neural Networks. In: 16th International Conference on Database and Expert Systems Applications (DEXA 2005), LCNS 3588, Springer-Verlag, Berlin/Heidelberg, pp. 85--94 (2005)
12. Clark, J., Koprinska, I., Poon, J.: A Neural Network Based Approach to Automated E-mail Classification. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI 2003) Halifax, Canada, Computer Society Press, pp. 702--705 (2003)
13. Pripužić, K., Huljenić, D., Carić A.: Vocabulary Development for Event Notification Services. In: Proceeding of The International Conference on Software, Telecommunications and Computer Networks SoftCOM 2004, Split-Dubrovnik-Venice, Croatia-Italy (2004)
14. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)