Sharif Spam Corpus: A Spam Filter Dataset for the Persian Language

Mojtaba Rohanian, Bahram Vazirnezhad, and Morteza Rohanian

Abstract—**Unsolicited bulk mail or spam are email messages that are unwanted or not requested by the recipient and are usually sent in large numbers. There are many ways to deal with spam but the most efficient ones are based on statistical analysis. Statistical techniques for e-mail filtering require the use of specific text corpora. A standard spam corpus should consist of hand-verified collections of emails classified into spam and ham categories. To our knowledge there is no publicly available spam corpus for the Persian language. In this project we built the first version of our Persian spam corpus and evaluated its effectiveness by randomly selecting subsets from the corpus, applying Naïve Bayes classification and assessing the results in terms of repeatability and precision. The tests indicate that the corpus could be reliably used to develop high precision statistical spam filters.**

***Index Terms*— Unsolicited bulk mail, spam corpus, Persian language, Naïve Bayes**

I. INTRODUCTION

Unwanted bulk mails have always been a problem for internet users as they waste resources and could pose potential security threats to the receiving systems. Spam detection is as of yet not fully solved and researchers still experiment with various anti-spam techniques. However the best and most practical anti-spam measures are those that utilize statistical methods.

Statistical spam filters need to be trained and tested using spam corpora. In this project we created a sizable spam corpus for the Persian language and tested the training corpus in a Naïve Bayes classification task to measure the system's consistency for achieving similar and highly accurate results across multiple tests.

A. Challenges

The spam corpus needs to reflect the contemporary Persian language, be comprehensive enough to capture obvious patterns, and encompass a wide range of subjects to be representative of the spam mails an average user actually receives.

Persian language is written using a modified Arabic alphabet, and sometimes transliterated in Latin alphabet on the internet. There is always need for normalization of text to have a homogenous style.

In preparing the ham messages, there is concern for respecting privacy of the correspondents.

II. RELATED WORK

There are several publicly available spam corpora for the English language. Some relate to a specific platform like cellphone messages [1] but others target the World Wide Web. The SpamAssassin Public Corpus [2] and 2007 TREC Public Spam Corpus [3] are two of the more prominent ones. Some are preprocessed, like the Enron-Spam while some might contain malicious code snippets and should be used with care [4].

The size of these corpora changes over a wide range. The PUI corpus consists of 1,099 messages, SpamBase contains 4, 601 with the majority being of the type ham and some like the TREC corpus number in tens of thousands [3].

The kind of processing also differs from one corpus to another. In the PUI corpus, HTML tags and attachments are removed. Fields other than *Subject* are deleted to ensure privacy. Words with frequencies smaller than a certain threshold are dropped. SpamBase represents the words using 58-dimensional arrays and puts aside the original messages [3].

The problem with vector representation is that it limits the researcher to the information contained in the vector, whereas a raw corpus that's carefully verified to be devoid of malicious code contains more information and can be used for a wider range of experiments.

III. PREPARATION OF THE CORPUS

To construct our Persian spam corpus we had two primary sources of information:

a. Personal emails

b. Publicly available messages, comments, correspondences and emails

Personal emails contained spam and ham messages in their original form. The anonymity of the correspondents was

Mojtaba Rohanian is with Languages and Linguistics Center at Sharif University of Technology, Tehran, Iran (e-mail: rohanian@mehr.sharif.edu).

Bahram Vazirnezhad is with Languages and Linguistics Center at Sharif University of Technology, Tehran, Iran. (e-mail: bahram@sharif.edu).

Morteza Rohanian was with Tehran University of Art, Tehran, Iran (e-mail: mailrohanian@gmail.com).

ISBN: 978-0-9803267-7-2

preserved unless the writers gave their consent.

To use the other sources, Python scripts were written to scrape the relevant pages and in some cases text was extracted manually. These sources include correspondences between famous people, public letters, publicly available email correspondences, and discussions in public forums that also included commercial spam messages. The HTML code was removed later. We ended up with 3000 messages containing equal number of hams and spams that were meticulously classified by hand. Those spam messages that mostly included images and URLs were kept intact.

To normalize the corpus, we used the Python NLP toolkit *Hazm* for the Persian language which provides a reliable normalizer and spelling corrector [5].

IV. ASSESSMENT IN A CLASSIFICATION TASK

In the rest of the paper we will explain in detail how we tried to test the corpus by training a Naïve Bayes classifier on randomly selected subsets of it. Then we evaluate the results.

A. Naïve Bayes Classification

Naïve Bayes is an established statistical method in machine learning that since late 1990's and after pioneering work by Sahami et al [6][7] has also been applied to spam detection. This method simply applies the Bayes Theorem on the corpus and assumes that the probability of words are independent from one another. This means that the classifier ignores word order and syntactic dependencies and thus could be called a bag of words model. Nevertheless it has been successfully combined with rule-based syntactic methods as well [8].

In spam classification, it is important to note that *false positives* are more undesirable than false negatives. I.e., the user would rather see a spam message in her inbox than lose an important email that is falsely designated as spam.

Assuming that a message contains n words (E = $w_1, \ldots, w_n$), the probability of E being an spam is:

$$\mathrm{P(E)} = \mathrm{P}(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i) \tag{1}$$

If we take S and H to denote spam and ham respectively, we need to compute P(H|E) and P(H|S). These are not directly computable but we have access to their inverse probabilities:

$$\mathrm{P(E}|S) = \mathrm{P}(w_1, \ldots, w_n|S) = \prod_{i=1}^{n} P(w_i|S)$$

$$\mathrm{P(E}|H) = \mathrm{P}(w_1, \ldots, w_n|H) = \prod_{i=1}^{n} P(w_i|H)$$

To compute individual probabilities we need to use the Bayes theorem:

$$\mathrm{P}(w_i|S) = \frac{P(w_i \cap S)}{p(S)} \tag{2}$$

To do this, we would need to divide the number of times w_i has occurred by the total number of words in both the ham and spam training sets. Applying the Bayes Theorem would give us the desired probabilities [9].

$$\mathrm{P}(S|E) = \frac{P(E|S)P(S)}{p(E)} = \frac{P(S)\prod_{i=1}^{n} P(w_i|S)}{p(E)} \tag{3}$$

$$\mathrm{P}(H|E) = \frac{P(E|H)P(H)}{p(E)} = \frac{P(H)\prod_{i=1}^{n} P(w_i|H)}{p(E)} \tag{4}$$

B. Paul Graham's modifications

Paul Graham has introduced a modified version of Naïve Bayes that has some simplifying assumptions but has proved to be reliably accurate in practice [10, 11]. Here, we decided to use his version, as it simplified computation. Paul Graham assumes the prior possibilities of P(S) and P(H) to be the same.

In traditional Naïve Bayes we have $\mathrm{P}(S|E) = \frac{P(E|S)P(S)}{p(E)}$ which based on the Bayes theorem is equivalent to:

$$\frac{P(E|S)P(S)}{\mathrm{P(S)P(E|S)} + \mathrm{P(H)P(E|H)}}$$

Assuming the probabilities of words are independent it follows:

$$\mathrm{P}(S|E) = \frac{P(S)\prod_{i=1}^{n} P(w_i|S)}{P(S)\prod_{i=1}^{n} P(w_i|S) + P(H)\prod_{i=1}^{n} P(w_i|H)} \tag{5}$$

$$\mathrm{P}(w_i|S) = \frac{P(S|w_i)P(w_i)}{p(S)} \tag{6}$$

Combing the two would result in the following formula:

$$\mathrm{P(S|E)} = \frac{\mathrm{P(S)}^{1-\mathrm{n}}\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(S|w_i)}}{\mathrm{P(S)}^{1-\mathrm{n}}\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(S|w_i)} + \mathrm{P(H)}^{1-\mathrm{n}}\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(H|w_i)}} \tag{7}$$

With Graham's assumption we could simplify the equation:

$$\mathrm{P(S|E)} = \frac{\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(S|w_i)}}{\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(S|w_i)} + \prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(H|w_i)}}$$

Graham also assumes that $\mathrm{P(H|w_i)}$= 1 - $\mathrm{P(S|w_i)}$ which will eventually lead to this final formula [10]:

$$P = \frac{p_1 p_2 p_3 \ldots p_n}{p_1 p_2 p_3 \ldots p_n + (1-p_1) + (1-p_2) + \cdots (1-p_n)} \quad (8)$$

The feature selection is done in a way that the resulting probabilities are either near 0 or close to 1. If the probability is over 0.9 it is ruled to be spam.

Words that have frequencies of less than 5 in both spam and ham training sets get ignored. For the words that have occurred in only the spam training set, if the frequency is over 10, the probability is set to be 0.9999, otherwise 0.9998. A similar process is done when dealing with words that happen only in the ham corpus. Words that are seen for the first time get the probability of 0.4, so they won't change the final combined probability too much. [10]. This could potentially allow spammers to do dictionary attacks by stuffing emails with neutral words to balance out the combined probability. To avoid this, Graham uses the concept of *interestingness* to limit the number of analyzed words to only the first 15 "interesting" words. The interestingness of a word is defined to be its absolute difference from the neutral probability 0.5. With this ranking method, only the 15 words that have probabilities farther from the center are considered. That's why the combined probabilities are always near 0 or 1 rather than in the middle.

As false positives are to be avoided, Graham introduced another modification. When computing individual probabilities for each word, we double the number of occurrences of words in the ham corpus. Meaning, if NS is the frequency of a word in the spam corpus, TS is the total number of spam mails, AH is the frequency of the word in the ham corpus and TH is the total number of ham mails, we have the following:

$$P(S|w_i) = \frac{\frac{NS}{TS}}{\frac{NS}{TS} + 2\left(\frac{AH}{TH}\right)} \quad (9)$$

Paul Graham's method is distinctive in that, unlike most other classifications, it doesn't use a stemmer in the preprocessing phase and words get recorded in their original form. This is an advantage, because it prevents the spammer from sidestepping the filters by intentionally misspelling problematic words [7].

C. Applying Naïve Bayes on the Sharif corpus

In order to test the Persian spam corpus with this classification method, we randomly selected 500 spam and 500 ham mails. From each of these two sets, 300 were used for training and the remaining 200 were used for testing. We repeated the exact same experiment 3 times with different randomly selected data.

To evaluate the system, we use precision, recall and f-measure which are defined as the following [12]:

$$\text{precision} = \frac{true\ positives}{true\ positives + false\ positives} \quad (10)$$

$$\text{recall} = \frac{true\ positives}{true\ positives + false\ negative} \quad (11)$$

$$F = 2\left(\frac{precision\ .\ recall}{precision + recall}\right) \quad (12)$$

In each of the three experiments, the system had access to 300 hams and 300 spams for training and was tested on 400 mails. On average over multiple tests, out of 200 spams, the system has misidentified 15 spams as hams. Out of 200 hams, the average number of false positives was 3. The worst case precision was 4 false positives and the best was 2. Therefore precision, recall and f-measure are 98.5%, 92.5% and 95.5% respectively.

In such a system, precision is much more important than recall, because false positives are more costly. In cases like this where precision and recall don't have an equal weight we could use a different measure called F_b [13].

$$F_b = (1+b^2) \cdot \frac{precision\ .\ recall}{(b^2.\ precision) + recall} \quad (13)$$

We take b = 0.5, which means that recall is 0.5 times more important than precision. That would result in $F_{0.5} = 97.25$.

V. CONCLUSION

The results in the previous section show that the corpus seems to be reliable and consistent enough to create high precision spam filters with good repeatability. The Sharif spam corpus currently holds 3000 classified emails. This will hopefully increase in the next versions to function as a freely available spam corpus to be used by researchers interested in processing of text in the Persian language.

REFERENCES

[1] Gómez Hidalgo, J.M., Cajigas Bringas, G., Puertas Sanz, E., Carrero García, F. "Content Based SMS Spam Filtering." Proceedings of the 2006 ACM Symposium on Document Engineering (ACM DOCENG'06), Amsterdam, The Netherlands, 10-13, 2006. SMS spam collecion avaialble at: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

[2] The SpamAssassin Public corpus is available at: http://spamassassin.apache.org/publiccorpus/

[3] The 2007 TREC Public Spam Corpus is available at: http://plg.uwaterloo.ca/~gvcormac/treccorpus07/

[4] Enrique Puertas Sanz, José María Gómez Hidalgo, José Carlos Cortizo, "Email Spam Filtering" in Advances in Computers, vol. 74, 2008, p. 45-114

[5] Hazm Python library available at: http://www.sobhe.ir/hazm/

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

[7] Patrick Pantel, Dekang Lin, “SpamCop -- A Spam Classification & Organization Program”, Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998

[8] Bill Yerazunis, “Sparse Binary Polynomial Hash Message Filtering and The CRM114 Discriminator”, Proceedings of 2003 Spam Conference, 2003

[9] Tianhao Sun “Spam Filtering based on Naive Bayes Classification”. Archive of research papers at Babes Bolyai University, 2009. Retrieved 2014-09-20 from:
http://www.cs.ubbcluj.ro/~gabis/DocDiplome/Bayesian/000539771r.pdf

[10] Paul Graham, “Better Bayesian Filtering”. In proceedings of the 2003 Spam Conference, 2002.
Available at: http://www.paulgraham.com/better.html

[11] Paul Graham, “A Plan for Spam”, in Hackers and Painters, Big Ideas from the Computer Age, O'Reilly, 2004, p. 121

[12] David L Olson, Dursun Delen, Advanced Data Mining Techniques, Springer, 1st edition February 1, 2008, p. 138

[13] C. J Van Rijsbergen, Information Retrieval, Butterworth, 2n edition, 1979