

Spam detection with PHP

Cesar D. Rodas

cesar@sixdegress.com.br

www.thyphp.com

Centro Nacional de Computación

Campus de la UNA - 2160

San Lorenzo - Paraguay

Introduction

Day by day, with the exponential growth of the Internet it is harder to classify useful information from spam. Spam is unsolicited or undesired bulk electronic messages, it could be in electronic emails, forums, mobile phone messages, newsgroups, blogs, and in several others places.

Because the PHP is the widest web development language, we propose an novel solution to classify spam from what is not, written 100% in PHP. The solution is the implementation of the [Bayesian spam filtering](#)

How does this work?

The [Bayesian spam filtering](#) is the process of using a [Naive Bayes classifier](#) to identify text spam. This technique is used by [SpamAssassin](#), [SpamBayes](#), [Bogofilter](#) and [ASSP](#).

All the system could be reduced to a simple mathematic expression, which is [Bayes' theorem](#) which returns the probability that an text is spam, given that it has certain words in it, is equal to the probability of finding those certain words in spam email, times the probability that any email is spam, divided by the probability of finding those words in any email:.

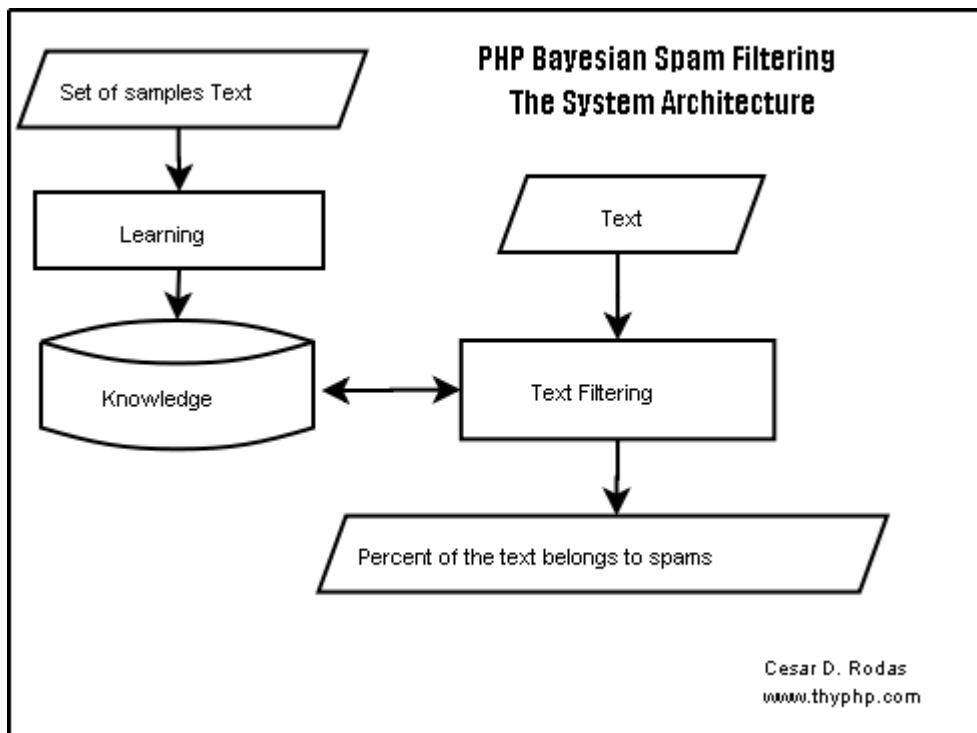
Suppose we have to know what is the probability of a given text to be a spam message or not if it has the word "Phentermine". Suppose that "Phentermine" occur 5 times in true text and 800 on spam, and we have a set of 100 true texts and 1000 spam.

$$P(\text{Phentermine}, \text{Spam}) = \frac{\frac{800}{100}}{\left(\frac{5}{100} + \frac{800}{1000}\right)} = 0,99$$

Also here comes another problem, the process of toke a text into word is different for every language, and also there is many differences forms of a work for every language, example: walk, walking, walked, caminar, caminó, caminando. All those words represents the same meaning but it has another written form, and therefore a difference spam score.

To solve that problem we implement a [N-gram](#)(is a sub-sequence of n items from a given sequence) based text instead of words. That is because in all the language words are

sequences of letters.



In the upper graphic you can see how the system works, basically the system must have a set of texts, which are manually classified as spam or not.

How implement

To implement the Bayesian Spam Filter for PHP you must download it from here <http://cesars.users.phpclasses.org/browse/package/4236.html> for free. The project is protected under the BSD license, which made you free to include the source in closed projects.

The system comes with not spam rules, it must learn first, also this class provides only the computation method, you must write your own method to save what is learned by the system or use the example which saves what is learned into a MySQL database.

The Learning phase is describe by the follow formula

$$count_{(ngram, set)} = \sum_{ngram \in set} 1$$

$$accuracy_{(ngram)} = \frac{count_{(ngram, spam)}}{(count_{(ngram, spam)} + count_{(ngram, notspam)})}$$

By this formula we have a mathematic method to determine the accuracy of a text belong to spam by its n-grams.

$$isSpam = \frac{(\sum_{ngram \in Spam} accuracy_{ngram})}{i} \times 100$$

Improvements to the system and applications

Because we are in the Web 2.0 era, where users can rate content and also categorize, an interesting project could be categorize using this class and also give the possibility to users rate as spam or not. If the system classified bad, it could learn it again. Then the site will always filtering almost any kind of spam content, even if spammers change the way of theirs spam.

Because the unwanted content from a site could be desire to a site the class is offer also a personalized concept of what is "spam".

Future work

We want this project growths up with the peoples contribution and suggestion at the [class forum](#). Also peoples can contact me in my email to suggest, ask, tell its experience, and anything related.