Email Filters that use Spammy Words Only

Vasanth Elavarasan
Department of Computer Science
University of Texas at Austin

Advisors: Mohamed Gouda
Department of Computer Science
University of Texas at Austin

12 May, 2006

Abstract

There are several types of email filters that can be used to classify emails into spam emails or non-spam emails. All these filters use the occurrence of spammy words (those words that are typically found in spam emails) and non-spammy words (those words that are typically found in non-spam emails) to compute the probability that a given email is spam. Some spam emails imitate non-spam emails by including passages on an unrelated subject matter. To solve this problem, we design an email filter using only spammy words. The success of an email filter is highly dependent on which words are used as the spammy words. We go on to define a method to identify the optimal set of spammy words to use in our filter.

Table of Contents

1. Introduction

The fight against spam has dramatically changed over the last decade. Today, spammers face greater difficulty in sending spam emails. A spammer must usually pass various checks to deliver an email. The spammer can rarely guarantee that a spam email has been delivered to a recipient. These obstacles have thwarted the delivery of spam emails and have allowed the overall frequency of spam to decline. However, it is inevitable that spammers will adapt their methods to bypass these checks. Internet service providers and network administrators use various techniques to filter emails. These techniques range from network level filters to complex machine learning filters. We focus our attention to content based statistical filters.

Content based filters classify an email as spam or non-spam purely on the content of the email. Given an email, they compute the probability that it is spam based on the occurrence of words within the email [1] [6]. Unfortunately, during our research, we found that the variety in the implementation of these email filters results in different filter rates. Therefore, we formally define an email filter and how it is implemented. Current literature constructs email filters such that the set of discriminating words include spammy words (those words that are typically found in spam emails) and non-spammy words (those words that are typically found in non-spam emails). We have observed that including non-spammy words in the set allows some spam emails containing these words to be classified as non-spam. To avoid using filters with this weakness, we construct our filters by using only spammy words. We introduce a method to define what words are considered spammy and how to identify the set of words that will result in the most optimal results.

2. Spammy Words

Let U be the set of all possible *emails*. For convenience, we assume that U is finite. Each email in U is a sequence of words. The emails in U are partitioned into two subsets: *spam* and *non-spam*.

We adopt the following notation in expressing probabilities over the emails in U.

- P(spam) denotes the probability that a given email in U is a spam email. In other words,

$$P(spam) = \frac{\#\, spam\, emails\, in U}{\#\, emails\ \ in U}$$

- P(w) denotes the probability that a given email in U has the word w. Thus,

$$P(w) = \frac{\#\, emails\, that\, contain\, w\ \ in U}{\#\, emails\, in U}$$

- We follow the well-established tradition of using "," to denote the "and" connector. Thus, $P(spam, w)$ denotes the probability that a given email in U is a spam email "and" has the word w. Note that the $P(spam, w) = P(w, spam)$. Also, $P(w_1, w_2)$ denotes the probability that a given email in U has the two words w_1 "and" w_2.

- We adopt the symbol "$\overline{}$" to denote the negation operator. For example, $P(\overline{spam}, w)$ denotes the probability that a given email in U is non-spam and has the word w. Also, $P(w_1, \overline{w_2})$ denotes the probability that a given email in U has the words w_1 but not the word w_2.

- $P(w \mid \overline{spam})$ denotes the conditional probability that a given email in U has the word w under the assumption that this email is non-spam.

  From the theory of probability, we have:

$$P(w \mid \overline{spam}) = \frac{P(w, \overline{spam})}{P(\overline{spam})}$$

A word w is called *spammy* if and only if it satisfies the following two conditions:

i. *Alpha Condition :*

   $P(w \mid spam) \geq \alpha \cdot P(w \mid \overline{spam})$, where α is some constant in the range 1.0 – 1.5.

ii. *Beta Condition :*

   $P(w \mid spam) > \beta$, where β is some constant in the range 0.1 – 0.7.

The Alpha condition implies that the occurrence of a spammy word in an email is an indicator that the email is spam. This is established in the following theorem.

Theorem 1:

For any spammy word w,

$$P(spam \mid w) \geq P(\overline{spam} \mid w)$$

provided that $P(spam) \geq P(\overline{spam})$

Proof:

$P(spam \mid w) =$ {from Bayes' Theorem [5] }

$$\frac{P(spam)P(w \mid spam)}{P(w)}$$

$\geq$ {from Alpha condition}

$$\frac{P(spam)P(w \mid \overline{spam})}{P(w)}$$

$\geq$ {from $P(spam) \geq P(\overline{spam})$ }

$$\frac{P(\overline{spam})P(w \mid \overline{spam})}{P(w)}$$

$= \quad P(\overline{spam} \mid w)$ □

Note that Theorem 1 is based on the assumption that $P(spam) \geq P(\overline{spam})$. This assumption is consistent with practical situations where $P(spam)$ is four times or more than $P(\overline{spam})$.

The Beta condition is intended to ensure that any word w that does not occur in any (spam or non-spam) email is not admitted as a spammy word even though it trivially satisfies the Alpha condition (since $P(w \mid spam) = P(w \mid \overline{spam}) = 0$).

3. A Filter Using One Spammy Word

Let w be a spammy word (that occurs in some emails in U). From Theorem 1, we have $P(spam \mid w) \geq P(\overline{spam} \mid w)$. This theorem suggests that we may use w by itself to identify whether a given email is spam or non-spam as follows:

If an email has w
    then this email is spam
    else this email is not spam

The "then" part of this filter seems reasonable in the light of Theorem 1. However, the "else" part of this filter can be wrong because the email may have another spammy word w' that may increase the conditional probability of spam, namely $P(spam \mid \overline{w}, w')$, beyond the conditional probability of non-spam, namely $P(\overline{spam} \mid \overline{w}, w')$. This is illustrated by the following theorem.

Theorem 2:

For every pair of distinct spammy words w and w', where

$$\frac{P(\overline{w} \mid \overline{spam})}{P(\overline{w} \mid spam)} \leq \frac{P(w' \mid spam)}{P(w' \mid \overline{spam})} \qquad (*)$$

we have

$$P(spam \mid \overline{w}, w') \geq P(\overline{spam} \mid \overline{w}, w')$$

provided that $P(spam) \geq P(\overline{spam})$.

Proof:

$$P(spam \mid \overline{w}, w') = \quad \{\text{from Bayes' Theorem}\}$$

$$\frac{P(spam)P(\overline{w}, w' \mid spam)}{P(\overline{w}, w')}$$

$$\geq \ \{\text{from independence of w and } w'\}$$

$$\frac{P(spam)P(\overline{w} \mid spam)P(w' \mid spam)}{P(\overline{w}, w')}$$

$$\geq \ \{\text{from}(*)\}$$

$$\frac{P(spam)P(\overline{w} \mid \overline{spam})P(w' \mid \overline{spam})}{P(\overline{w}, w')}$$

$$\geq \ \{\text{from } P(spam) \geq P(\overline{spam})\}$$

$$\frac{P(\overline{spam})P(\overline{w} \mid \overline{spam})P(w' \mid \overline{spam})}{P(\overline{w}, w')}$$

$$= \ \{\text{from independence of } \overline{w} \text{ and } w' \text{ [2]}\}$$

$$\frac{P(\overline{spam})P(\overline{w}, w' \mid \overline{spam})}{P(\overline{w}, w')}$$

$$\geq \{\text{from Bayes' Theorem}\}$$

$$P(\overline{spam} \mid \overline{w}, w') \qquad \square$$

From Theorem 2, if one is to use some spammy words to construct an email filter, then one should use all the spammy words in U, or at least all those that occur in the training set, to construct their email filters.

4. A Filter Using All Spammy Words

In this section, we describe how to compose an email filter that uses all the spammy words in a training set T, where T is a given set of spam and non-spam emails that is a "good representative" of the set of all emails U. The three steps to compose the filter are as follows.

Step 1: Choose a value for α from the domain, 1.1 through 1.5, of all α values. Also choose a value for β from the domain, .01 through .07, of all β values.

Step 2: Use the chosen values of α and β to identify every spammy word w in the training set T. Each identified spammy word needs to occur in some email in T and to satisfy the Alpha and Beta conditions discussed in Section 2. Let the identified spammy words be w_1, w_2, ,..., w_n.

Step 3: Use the training set T to compute $P(spam)$ and $P(\overline{spam})$. (Note that $P(\overline{spam}) = 1 - P(spam)$.) Also, use T to compute for every spammy word w_i, identified in Step 2, the following four probabilities:

$$P(w_i \mid spam) \;,\; P(\overline{w_i} \mid spam),$$

$$P(w_i \mid \overline{spam}) \;,\; P(\overline{w_i} \mid \overline{spam})$$

(Note that $P(\overline{w_i} \mid spam) = 1 - P(w_i \mid spam)$ and that $P(\overline{w_i} \mid \overline{spam}) =$ $1 - P(w_i \mid \overline{spam})$.)

Now, we can use the composed filter to classify any given email as spam or non-spam. Without a loss of generality, assume that the given email has the spammy words $w_1,...,w_m$ but does not have the spammy words $w_{m+1}, ,...,w_n$. Classification of the given email consists of computing the probability

$$P(spam \mid w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n})$$

If this computed probability is ≥ .5, then we conclude that the given email is spam. Otherwise, the given email is non-spam.

In order to compute the probability $P(spam \mid w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n})$, we use Bayes' Theorem and the independence assumption as follows

$$\begin{aligned} &P(spam \mid w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n}) \\ &\qquad = \{\text{from Bayes' Theorem}\} \\ &\qquad \frac{P(spam)P(w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n} \mid spam)}{P(w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n})} \\ &\qquad = \{\text{from the independence assumption}\} \\ &\qquad \frac{P(spam) \prod_{i=1}^{m} P(w_i \mid spam) \prod_{i=m+1}^{n} P(\overline{w_i} \mid spam)}{P(w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n})} \end{aligned}$$

The numerator of the last expression can be easily computed using the quantities $P(spam)$, $P(w_i \mid spam)$, and $P(\overline{w_i} \mid spam)$ which are computed in the filter from the training set T. It remains now to compute the denominator, $P(w_1,..., w_m, \overline{w_{m+1}},..., \overline{w_n})$, of the last expression.

To compute the denominator $P(w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n})$, we resort to the well-known equation:

$$P(spam \mid w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n}) + P(\overline{spam} \mid w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n}) \quad = 1$$

It is straightforward to show from this equation, using Bayes' Theorem and the independence assumption, that

$$\begin{aligned} &P(w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n}) \\ &\qquad = P(spam) \prod_{i=1}^{m} P(w_i \mid spam) \prod_{i=m+1}^{n} P(\overline{w_i} \mid spam) \\ &\qquad + \\ &\qquad P(\overline{spam}) \prod_{i=1}^{m} P(w_i \mid \overline{spam}) \prod_{i=m+1}^{n} P(\overline{w_i} \mid \overline{spam}) \end{aligned}$$

Therefore, the denominator, $P(w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n})$, can also be computed using the quantities that are compiled in the filter the from training set T.

In summary, the probability $P(spam \mid w_1,...,w_m,\overline{w_{m+1}},...,\overline{w_n})$ can be computed for any given mail (that has the spammy words $w_1,...,w_m$ but does not have the spammy words w_{m+1}, ,...,w_n) using the quantities that are computed in the filter from the training set T.

5. Computing α_{max} and β_{max}

In the previous section, we showed that one can choose a value for α and a value for β, and use these chosen values along with a given training set T to compose a filter, denoted f(α**,** β), that can be used to classify emails into spam and non-spam. In this section, we show how one can use the given training set T, to search for a particular combination of α value and β value, denoted (α_{max} , β_{max})

such that the effectiveness of the filter f(α_{max}, β_{max}) is not less than the effectiveness of any other filter f(α, β). The search procedures for the (α_{max}, β_{max}) combination can be specified as follows.

For every combination of (α, β) of an α value and β value

do

i. use (α, β) and the given training set T to compute a filter f(α, β) as discussed in Section 4.

ii. use filter f(α, β) computed in i to classify all the emails in the given training set T into spam and non-spam. (This classification can prove correct for some emails and wrong for the others.)

iii. Let SS(α, β) denote the percent of spam emails in T, that are correctly classified as spam by the filter f(α, β). Also, let NS(α, β) denote the percent of non-spam emails in T that are wrongly classified as spam by the filter f(α, β).

od

The combination (α_{max}, β_{max}) is the one whose filter f(α_{max}, β_{max}) yields the highest value of

$$SS(\alpha_{max}, \beta_{max}) - NS(\alpha_{max}, \beta_{max})$$

In other words, for every (α, β) combination, we have

$$[SS(\alpha_{max}, \beta_{max}) - NS(\alpha_{max}, \beta_{max})] \geq [SS(\alpha, \beta) - NS(\alpha, \beta)]$$

6. Experimental Results

To validate our approach, we implemented a filter using all spammy words. The goal was to show that email filters using only spammy words is as successful if not better than email filters using spammy and non-spammy words. We also intended to show that our algorithm for determining α and β would create the best performing filter.

The Ling-Spam email corpus was used to generate a training set T of emails. The corpus is a mixture of 481 spam messages and 2412 messages sent via the Linguist list (Linguist List). Attachments, HTML tags, and duplicate spam messages received on the same day are not included. This corpus is the same as the one described in [1]. We partitioned the set of messages into a training set of 432 spam emails and 2170 non-spam emails. The remaining 49 spam emails and 242 non-spam emails are designated as the test set.

The email filter is constructed as explained in Section 4. The filter computes the probability that a given email is spam based on the occurrence of words in that email. The filter only considers those words that are defined to be spammy based on the conditions α and β. The algorithm determines the optimal values α_{max} and β_{max} by analyzing the filtering results obtained using different values for α and β.

We said α was in the range 1.0 to 1.5, and β was in the range from 0.01 to 0.07. Calculating SS(α , β) over the range of values of α and β using the training set results in the values shown in Table 1. Calculating NS(α , β) over the range of values of α and β using the training set results in the values shown in Table 2.

Table 1 : % of spam classified as spam SS(α , β)

β

α		0.01	0.02	0.03	0.04	0.05	0.06	0.07
	1	56.8	64.2	68.8	75	77.1	79.9	80.1
	1.1	66.5	69.3	72.5	76.2	79.2	78.5	77.4
	1.2	68.3	71.6	74.8	76.9	77.4	77.6	75
	1.3	70.4	72.5	74.8	77.1	77.4	77.6	74.8
	1.4	71.8	73.4	76	76.7	76.4	75.7	74.1
	1.5	73.7	73.7	75.7	76.9	76	75	73.7

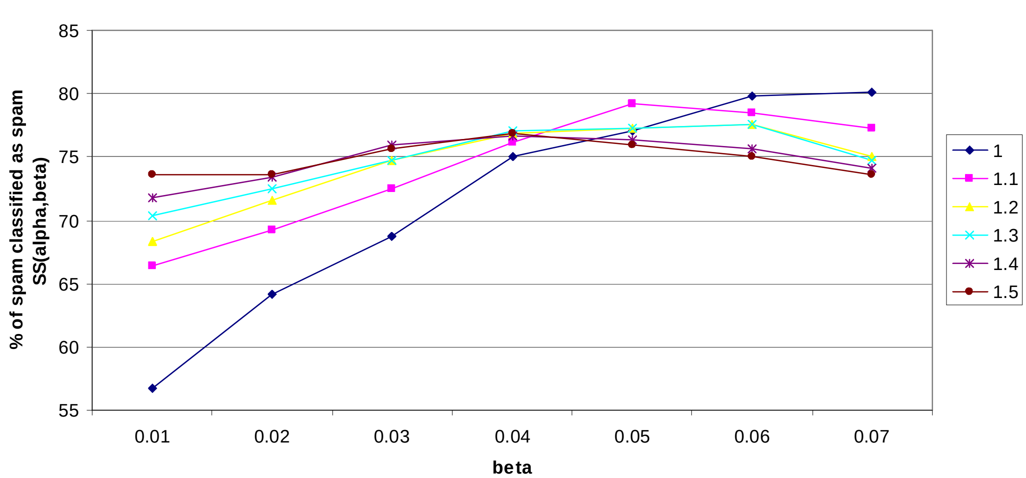
Table 2 : % of non-spam classified as spam NS(α , β)

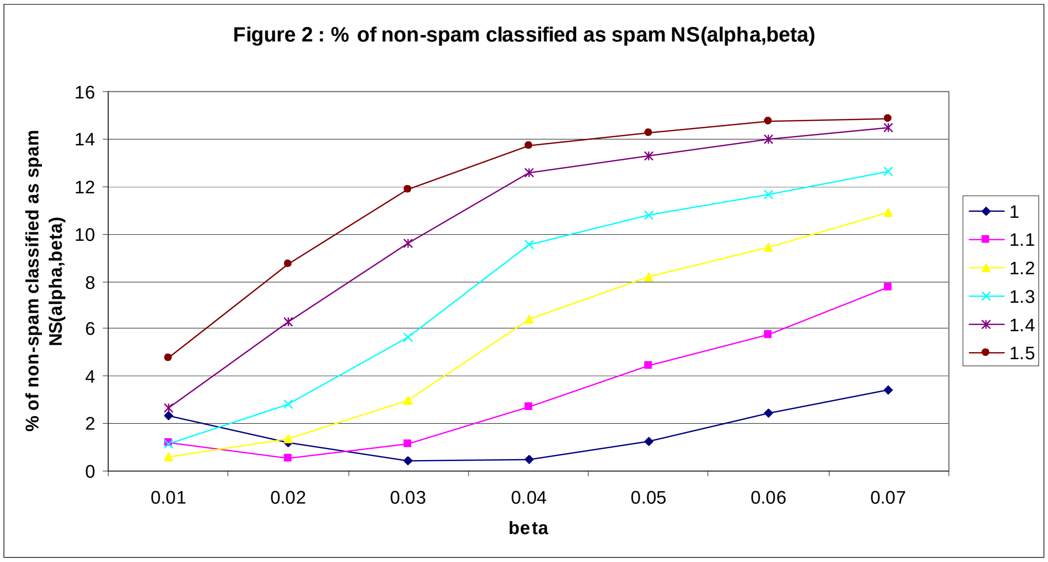
β

α		0.01	0.02	0.03	0.04	0.05	0.06	0.07
	1	2.4	1.2	0.5	0.6	1.3	2.5	3.5
	1.1	1.2	0.6	1.2	2.8	4.5	5.8	7.8
	1.2	0.6	1.4	3	6.5	8.3	9.5	11
	1.3	1.2	2.9	5.7	9.6	10.8	11.7	12.7
	1.4	2.7	6.3	9.6	12.6	13.3	14.1	14.5
	1.5	4.8	8.8	11.9	13.8	14.3	14.8	14.9

The graphical representation of Table 1 (Figure 1) and Table 2 (Figure 2) are as follows.

Figure 1 : % of spam classified as spam SS(alpha,beta)





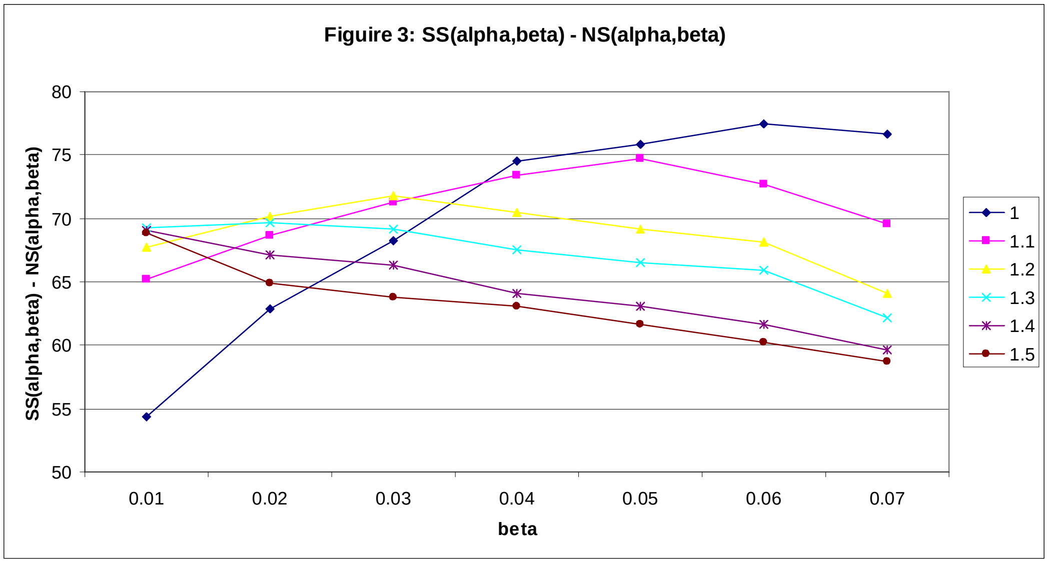
Calculating SS(α , β) - NS(α , β) over the range of values of α and β using the training set results in the values shown in Table 3.

Table 3 : SS(α , β) - NS(α , β)

β

α		**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**
	1	54.4	63	68.3	74.4	75.8	77.4	76.6
	1.1	65.3	68.7	71.3	73.4	74.7	72.7	69.6
	1.2	67.7	70.2	71.8	70.4	69.1	68.1	64
	1.3	69.2	69.6	69.1	67.5	66.6	65.9	62.1
	1.4	69.1	67.1	66.4	64.1	63.1	61.6	59.6
	1.5	68.9	64.9	63.8	63.1	61.7	60.2	58.8

The graphical representation of Table 3 is as follows.



Since α = 1.0 and β = 0.06 result in the highest value in the table, we set α_{max} to 1.0 and β_{max} to 0.06. In the final step of the algorithm, we apply the filter using α_{max} and β_{max} to our test set to see what results we obtain. Using α_{max} and β_{max} on the test set of emails results in SS(α_{max}, β_{max}) equaling 75.5% and NS(α_{max}, β_{max}) equally 3.3%. There will always be some discrepancy between SS(α_{max}, β_{max}) and NS(α_{max}, β_{max}) of the training set and test set since both sets consist of different emails. We believe a 4.4% difference in SS(α_{max}, β_{max}) and a 0.8% difference in NS(α_{max}, β_{max}) is tolerable and validates that α_{max} and β_{max} are acceptable values. The set of words used for this filter can be found in the Appendix.

Our experimental results have verified that filtering emails based on spammy words is as successful as filtering on occurrences of all words. We benchmarked our success by comparing our results with other literature. Both [1] and [6] report spam detection rates from 60% to 85%, while non-spam classification errors from 0% to 8%. Current software filters like Bogofilter, SpamAssassin, and SpamBayes also report that on average they have spam rates above 80% and non-spam classification errors from 0% to10% [3] [7] [8]. The exact filter rates one achieves through the software is dependent on customizable options the email user enables. We believe the main reason for higher success rates in software filters is their use of features other than words alone, i.e. time of arrival, attachments, embedded html.

We also noticed that α and β determines the size of the set of words used to filter emails with. As α increases from 1.0 – 1.5 and β increases from 0.01 to 0.07, the set of N words decreases in size. All series of values for α over a changing β result in filter rates improving, and then decreasing after a certain point. These thresholds vary for each series of α and can not be determined without building filters over all possible ranges. The trend is different for non-spam rates as shown by Table B. As β increase, the non-spam classification error rate may decrease, but will immediately increase. Table 3 summarizes the experiment by showing that with the Ling-Spam corpus, the most successful filters are created using α as 1.0 or 1.1 and β between 0.05 and 0.07.

7. Concluding Remarks

Our research has shown one can achieve acceptable filter rates by using filters that only use spammy words. We have also defined a methodology for defining an email filter based on α and β. Our experiments have shown that computing over ranges of values for α and β and computing values SS(α, β) - NS(α, β) is the best method to identify which α and β should be used. Future research will be dedicated to improving the algorithm to construct the best filter. We intended to increase the granularity of ranges to see if that improves filter rates. We also want to gather more email corpuses and compare our algorithm to them. The hope is to discover a range of α and β that is consistent within all email corpuses. More improvements could be made to our filters by considering information other than words alone, such as attachments and embedded html. A technique we hope to pursue to further the fight against spam is to construct web filters. All spam emails refer the reader to a website or url. Our future filter will go to the website and classify the email as spam or non-spam based on the content of the webpage.

8. References

[1] Andortsopoulos, I., Koutsias, J., Chandrinos, K.V, Paliouras, G., and Spryopoulos, C.D. “An evaluation of naive Bayesian anti-spam filtering” In Proceeding of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17

[2] Bernardo, Jose M., and Adrian F. M. Smith. Bayesian Theory. England: Antony Rowe Ltd, 1994.

[3] BogoFilter, http://bogofilter.sourceforge.net/.

[4] Linguist List. http://listserv.linguistlist.org/archives/linguist.html. A moderated (hence, spam-free) list about the profession and science of linguistics.

[5] Nilsson, Nils J.. Artificial Intelligence: A New Synthesis. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1998.

[6] Sahami, M., Dumais, S., Heckerman D., and Horvits, E. (1998). “A Bayesian approach to filtering junk email”. In Learning for Text Categorization – Papers from the AAAI Workshop, pp. 55 – 62, Madison Wisconsin. AAAI Technical Report WS 98-05

[7] SpamAssassin, http://www.spamassassin.org/index.html.

[8] SpamBayes, http://spambayes.sourceforge.net/.

Appendix

This is the set of words used in the optimal filter calculated in Section 6 from the training set. Alpha is 1.0 and Beta is 0.06. Each row displays the word, number of spam emails with that word, number of non-spam emails with that word, and the P(word|spam). Note that our training set has a total of 432 spam emails and 2170 non-spam emails.

	Word	# of spam emails with word	# of non-spam emails with word	P(word|spam)
1	able	60	157	0.138888889
2	about	164	795	0.37962963
3	above	76	248	0.175925926
4	absolutely	43	19	0.099537037
5	accept	62	82	0.143518519
6	access	60	96	0.138888889
7	account	48	167	0.111111111
8	achieve	28	13	0.064814815
9	action	45	64	0.104166667
10	actually	34	101	0.078703704
11	ad	49	29	0.113425926
12	add	58	75	0.134259259
13	added	51	37	0.118055556
14	additional	71	137	0.164351852
15	address	190	613	0.439814815
16	addresses	89	148	0.206018519
17	ads	43	8	0.099537037
18	adult	53	29	0.122685185
19	advantage	50	31	0.115740741
20	advertise	46	2	0.106481481
21	advertisement	38	7	0.087962963
22	advertising	68	7	0.157407407
23	afford	26	8	0.060185185
24	after	103	269	0.238425926
25	again	102	135	0.236111111
26	against	41	81	0.094907407
27	age	33	53	0.076388889
28	ago	35	163	0.081018519
29	air	30	23	0.069444444
30	all	267	839	0.618055556
31	allow	42	81	0.097222222
32	allows	55	41	0.127314815
33	almost	41	61	0.094907407
34	alone	33	27	0.076388889
35	along	43	117	0.099537037
36	already	59	112	0.136574074

37	alter	26	7	0.060185185
38	always	78	115	0.180555556
39	am	85	347	0.196759259
40	amazing	59	2	0.136574074
41	america	30	115	0.069444444
42	amount	74	79	0.171296296
43	another	51	217	0.118055556
44	answer	50	79	0.115740741
45	any	199	735	0.460648148
46	anyone	70	252	0.162037037
47	anything	55	93	0.127314815
48	anywhere	66	25	0.152777778
49	aol	46	15	0.106481481
50	are	288	1412	0.666666667
51	around	42	156	0.097222222
52	ask	67	93	0.155092593
53	asked	35	145	0.081018519
54	attention	28	119	0.064814815
55	automatically	32	25	0.074074074
56	available	123	513	0.284722222
57	away	50	55	0.115740741
58	back	97	124	0.224537037
59	bank	59	54	0.136574074
60	be	310	1555	0.717592593
61	beach	27	10	0.0625
62	because	88	225	0.203703704
63	become	38	108	0.087962963
64	been	126	556	0.291666667
65	before	114	277	0.263888889
66	being	92	272	0.212962963
67	believe	71	114	0.164351852
68	below	126	334	0.291666667
69	benefits	27	27	0.0625
70	best	163	158	0.377314815
71	better	74	110	0.171296296
72	big	51	34	0.118055556
73	bill	30	51	0.069444444
74	bills	36	4	0.083333333
75	bottom	33	18	0.076388889
76	bought	29	9	0.06712963
77	box	71	182	0.164351852
78	brand	27	3	0.0625
79	break	38	103	0.087962963
80	build	40	30	0.092592593
81	bulk	73	11	0.168981481
82	business	140	67	0.324074074
83	businesses	37	1	0.085648148
84	buy	81	17	0.1875
85	ca	58	230	0.134259259
86	call	149	537	0.344907407

87	came	27	66	0.0625
88	campaign	27	2	0.0625
89	can	256	923	0.592592593
90	capital	48	24	0.111111111
91	car	37	35	0.085648148
92	card	82	106	0.189814815
93	cards	35	10	0.081018519
94	cash	85	18	0.196759259
95	cd	62	32	0.143518519
96	cents	27	2	0.0625
97	chance	45	35	0.104166667
98	change	60	184	0.138888889
99	changes	28	85	0.064814815
100	charge	48	49	0.111111111
101	check	133	114	0.30787037
102	checks	62	16	0.143518519
103	choice	30	69	0.069444444
104	choose	62	34	0.143518519
105	city	89	140	0.206018519
106	class	35	100	0.081018519
107	click	115	10	0.266203704
108	code	64	92	0.148148148
109	com	186	250	0.430555556
110	come	118	171	0.273148148
111	comes	45	70	0.104166667
112	coming	44	39	0.101851852
113	companies	47	18	0.108796296
114	company	97	39	0.224537037
115	competition	31	10	0.071759259
116	complete	59	137	0.136574074
117	completely	50	45	0.115740741
118	computer	78	279	0.180555556
119	connection	33	55	0.076388889
120	control	37	65	0.085648148
121	copy	69	255	0.159722222
122	corporation	32	27	0.074074074
123	cost	112	59	0.259259259
124	costs	45	47	0.104166667
125	could	100	308	0.231481481
126	country	44	82	0.101851852
127	course	48	215	0.111111111
128	created	31	29	0.071759259
129	credit	88	48	0.203703704
130	customers	49	3	0.113425926
131	cut	26	34	0.060185185
132	daily	27	23	0.0625
133	date	86	227	0.199074074
134	day	132	188	0.305555556
135	days	139	86	0.321759259
136	deal	34	78	0.078703704

137	dear	49	125	0.113425926
138	debt	27	5	0.0625
139	decide	46	20	0.106481481
140	decided	30	30	0.069444444
141	delete	41	10	0.094907407
142	delivery	43	13	0.099537037
143	designed	30	60	0.069444444
144	details	54	204	0.125
145	did	68	154	0.157407407
146	different	72	349	0.166666667
147	direct	44	123	0.101851852
148	directly	35	157	0.081018519
149	discover	36	17	0.083333333
150	do	253	566	0.585648148
151	does	91	399	0.210648148
152	doing	54	78	0.125
153	dollar	38	7	0.087962963
154	dollars	77	20	0.178240741
155	don	48	26	0.111111111
156	done	51	109	0.118055556
157	doubt	35	34	0.081018519
158	down	78	84	0.180555556
159	download	30	17	0.069444444
160	dreams	26	3	0.060185185
161	drive	28	42	0.064814815
162	each	121	334	0.280092593
163	earn	48	3	0.111111111
164	earth	26	14	0.060185185
165	easily	48	58	0.111111111
166	easy	120	57	0.277777778
167	effective	48	34	0.111111111
168	effort	49	44	0.113425926
169	else	62	89	0.143518519
170	email	148	488	0.342592593
171	emails	26	1	0.060185185
172	end	34	166	0.078703704
173	engines	32	5	0.074074074
174	enjoy	27	16	0.0625
175	enough	40	89	0.092592593
176	enter	43	17	0.099537037
177	entire	38	38	0.087962963
178	envelope	30	10	0.069444444
179	error	33	26	0.076388889
180	even	131	241	0.303240741
181	ever	99	74	0.229166667
182	every	136	110	0.314814815
183	everyone	56	54	0.12962963
184	everything	68	34	0.157407407
185	exactly	53	44	0.122685185
186	excellent	28	37	0.064814815

187	except	27	64	0.0625
188	excess	26	6	0.060185185
189	exciting	39	17	0.090277778
190	exclusive	27	25	0.0625
191	expect	29	64	0.06712963
192	experience	60	195	0.138888889
193	expiration	28	7	0.064814815
194	express	32	43	0.074074074
195	extra	48	43	0.111111111
196	extremely	30	39	0.069444444
197	fact	50	167	0.115740741
198	family	64	78	0.148148148
199	fast	35	29	0.081018519
200	faster	30	8	0.069444444
201	federal	34	13	0.078703704
202	fee	32	141	0.074074074
203	feel	34	70	0.078703704
204	few	80	193	0.185185185
205	file	52	96	0.12037037
206	files	38	75	0.087962963
207	fill	55	42	0.127314815
208	filled	27	20	0.0625
209	finally	38	78	0.087962963
210	financial	64	22	0.148148148
211	find	113	245	0.261574074
212	first	133	581	0.30787037
213	follow	64	108	0.148148148
214	following	122	573	0.282407407
215	for	378	1829	0.875
216	forget	28	13	0.064814815
217	form	79	387	0.18287037
218	found	59	247	0.136574074
219	four	48	188	0.111111111
220	free	253	137	0.585648148
221	freedom	45	12	0.104166667
222	fresh	46	7	0.106481481
223	friend	44	26	0.101851852
224	friends	61	26	0.141203704
225	from	273	1218	0.631944444
226	front	28	36	0.064814815
227	full	80	267	0.185185185
228	fully	27	68	0.0625
229	fun	59	9	0.136574074
230	future	82	125	0.189814815
231	games	29	9	0.06712963
232	generate	35	18	0.081018519
233	get	211	235	0.488425926
234	gets	31	19	0.071759259
235	getting	39	43	0.090277778
236	girls	26	8	0.060185185

237	give	110	173	0.25462963
238	giving	29	40	0.06712963
239	go	115	130	0.266203704
240	goal	37	58	0.085648148
241	going	60	110	0.138888889
242	gold	26	11	0.060185185
243	good	91	199	0.210648148
244	got	58	76	0.134259259
245	gov	26	18	0.060185185
246	great	96	138	0.222222222
247	greatest	31	10	0.071759259
248	growing	32	43	0.074074074
249	guarantee	65	8	0.150462963
250	guaranteed	48	6	0.111111111
251	guide	32	47	0.074074074
252	had	71	216	0.164351852
253	half	26	96	0.060185185
254	hand	48	104	0.111111111
255	happen	34	29	0.078703704
256	happy	38	64	0.087962963
257	hard	57	147	0.131944444
258	have	291	1008	0.673611111
259	having	36	121	0.083333333
260	hear	36	72	0.083333333
261	hello	47	16	0.108796296
262	help	94	218	0.217592593
263	her	38	118	0.087962963
264	here	185	305	0.428240741
265	hesitate	31	17	0.071759259
266	hi	33	39	0.076388889
267	high	61	119	0.141203704
268	highly	26	78	0.060185185
269	hit	36	14	0.083333333
270	home	121	175	0.280092593
271	hope	29	109	0.06712963
272	hot	41	13	0.094907407
273	hour	68	55	0.157407407
274	hours	81	54	0.1875
275	house	32	60	0.074074074
276	how	164	476	0.37962963
277	htm	28	51	0.064814815
278	http	178	734	0.412037037
279	huge	50	7	0.115740741
280	hundreds	85	7	0.196759259
281	id	29	16	0.06712963
282	idea	30	92	0.069444444
283	if	281	826	0.650462963
284	imagine	30	31	0.069444444
285	immediate	26	25	0.060185185
286	immediately	71	53	0.164351852

287	inc	32	107	0.074074074
288	include	89	438	0.206018519
289	included	41	147	0.094907407
290	includes	33	161	0.076388889
291	including	92	435	0.212962963
292	income	85	5	0.196759259
293	increase	39	30	0.090277778
294	industry	30	21	0.069444444
295	info	46	83	0.106481481
296	information	190	916	0.439814815
297	initial	34	78	0.078703704
298	instructions	82	44	0.189814815
299	internet	152	110	0.351851852
300	into	98	386	0.226851852
301	involved	28	100	0.064814815
302	is	335	1677	0.775462963
303	it	271	1039	0.627314815
304	job	42	97	0.097222222
305	join	45	43	0.104166667
306	just	207	253	0.479166667
307	keep	75	60	0.173611111
308	kind	32	138	0.074074074
309	knew	26	25	0.060185185
310	know	145	358	0.335648148
311	known	27	119	0.0625
312	large	42	136	0.097222222
313	last	69	204	0.159722222
314	later	41	162	0.094907407
315	latest	40	43	0.092592593
316	laws	38	15	0.087962963
317	learn	51	70	0.118055556
318	least	59	259	0.136574074
319	leave	42	49	0.097222222
320	legal	58	27	0.134259259
321	legitimate	34	16	0.078703704
322	less	75	160	0.173611111
323	let	83	170	0.19212963
324	letter	72	124	0.166666667
325	letters	35	78	0.081018519
326	level	47	173	0.108796296
327	life	78	90	0.180555556
328	like	165	561	0.381944444
329	limited	55	219	0.127314815
330	line	131	161	0.303240741
331	link	43	53	0.099537037
332	list	176	407	0.407407407
333	listed	45	91	0.104166667
334	lists	79	64	0.18287037
335	little	71	140	0.164351852
336	live	78	30	0.180555556

337	living	45	29	0.104166667
338	ll	126	92	0.291666667
339	local	31	153	0.071759259
340	long	50	224	0.115740741
341	look	61	143	0.141203704
342	looking	58	168	0.134259259
343	lose	48	9	0.111111111
344	lot	57	108	0.131944444
345	love	40	19	0.092592593
346	low	42	63	0.097222222
347	luck	32	8	0.074074074
348	made	81	299	0.1875
349	mail	197	759	0.456018519
350	mailbox	26	5	0.060185185
351	mailed	34	49	0.078703704
352	mailing	160	89	0.37037037
353	mailings	48	2	0.111111111
354	mails	42	4	0.097222222
355	major	69	172	0.159722222
356	make	182	285	0.421296296
357	makes	31	86	0.071759259
358	making	88	117	0.203703704
359	many	155	428	0.358796296
360	market	80	18	0.185185185
361	marketing	91	21	0.210648148
362	mastercard	39	25	0.090277778
363	matter	46	102	0.106481481
364	me	113	447	0.261574074
365	meet	35	36	0.081018519
366	members	48	167	0.111111111
367	message	117	187	0.270833333
368	method	41	90	0.094907407
369	million	80	18	0.185185185
370	millions	54	5	0.125
371	mind	37	89	0.085648148
372	minute	28	99	0.064814815
373	miss	34	13	0.078703704
374	money	169	58	0.391203704
375	month	80	52	0.185185185
376	monthly	32	5	0.074074074
377	months	55	50	0.127314815
378	more	223	777	0.516203704
379	most	145	363	0.335648148
380	move	40	38	0.092592593
381	much	126	270	0.291666667
382	multi	40	84	0.092592593
383	must	94	353	0.217592593
384	my	131	420	0.303240741
385	myself	26	47	0.060185185
386	n	157	499	0.363425926

387	name	169	392	0.391203704
388	names	60	149	0.138888889
389	necessary	29	96	0.06712963
390	need	151	219	0.349537037
391	needed	43	52	0.099537037
392	net	79	64	0.18287037
393	never	91	108	0.210648148
394	new	225	661	0.520833333
395	news	35	37	0.081018519
396	next	88	108	0.203703704
397	no	217	611	0.502314815
398	nor	26	64	0.060185185
399	not	250	1038	0.578703704
400	note	66	244	0.152777778
401	nothing	64	58	0.148148148
402	now	221	337	0.511574074
403	number	127	482	0.293981481
404	numbers	32	101	0.074074074
405	ny	31	105	0.071759259
406	obviously	29	36	0.06712963
407	off	88	84	0.203703704
408	offer	125	80	0.289351852
409	offers	60	78	0.138888889
410	office	63	122	0.145833333
411	old	55	159	0.127314815
412	once	86	102	0.199074074
413	one	217	933	0.502314815
414	online	75	39	0.173611111
415	only	229	541	0.530092593
416	open	45	143	0.104166667
417	opportunities	33	31	0.076388889
418	opportunity	80	110	0.185185185
419	or	290	1424	0.671296296
420	order	160	316	0.37037037
421	ordering	53	54	0.122685185
422	orders	88	47	0.203703704
423	organization	26	88	0.060185185
424	others	69	191	0.159722222
425	our	260	367	0.601851852
426	out	209	444	0.483796296
427	outside	26	73	0.060185185
428	over	189	243	0.4375
429	overnight	34	2	0.078703704
430	own	106	207	0.24537037
431	package	56	23	0.12962963
432	paid	46	55	0.106481481
433	part	63	308	0.145833333
434	participate	38	53	0.087962963
435	partners	32	6	0.074074074
436	pass	37	35	0.085648148

437	past	46	138	0.106481481
438	pay	89	60	0.206018519
439	payable	45	62	0.104166667
440	paying	26	10	0.060185185
441	payment	39	74	0.090277778
442	people	146	347	0.337962963
443	per	99	155	0.229166667
444	perfectly	29	22	0.06712963
445	period	29	66	0.06712963
446	person	52	148	0.12037037
447	personal	78	92	0.180555556
448	phone	129	312	0.298611111
449	piece	32	23	0.074074074
450	place	89	286	0.206018519
451	plan	44	45	0.101851852
452	plans	34	22	0.078703704
453	play	34	53	0.078703704
454	please	236	774	0.546296296
455	plus	79	129	0.18287037
456	po	26	83	0.060185185
457	postal	40	82	0.092592593
458	potential	55	92	0.127314815
459	power	34	60	0.078703704
460	powerful	38	21	0.087962963
461	practically	28	6	0.064814815
462	price	78	90	0.180555556
463	prices	31	31	0.071759259
464	print	69	56	0.159722222
465	prior	27	45	0.0625
466	probably	29	93	0.06712963
467	problem	39	172	0.090277778
468	process	45	148	0.104166667
469	produce	33	37	0.076388889
470	product	94	21	0.217592593
471	products	76	28	0.175925926
472	professional	41	58	0.094907407
473	profit	30	9	0.069444444
474	profitable	38	3	0.087962963
475	program	104	349	0.240740741
476	programs	69	75	0.159722222
477	proof	34	29	0.078703704
478	proven	51	13	0.118055556
479	provide	57	260	0.131944444
480	public	30	82	0.069444444
481	purchase	71	18	0.164351852
482	put	75	111	0.173611111
483	quality	33	78	0.076388889
484	questions	88	304	0.203703704
485	quick	36	16	0.083333333
486	quickly	29	26	0.06712963

487	rate	47	77	0.108796296
488	rates	31	46	0.071759259
489	re	137	317	0.31712963
490	reach	33	48	0.076388889
491	read	92	138	0.212962963
492	reading	43	145	0.099537037
493	ready	48	137	0.111111111
494	real	69	120	0.159722222
495	really	78	127	0.180555556
496	reason	45	83	0.104166667
497	receive	171	98	0.395833333
498	received	75	256	0.173611111
499	receiving	50	20	0.115740741
500	released	28	10	0.064814815
501	remember	61	55	0.141203704
502	remove	152	7	0.351851852
503	removed	109	16	0.252314815
504	reply	96	68	0.222222222
505	report	54	71	0.125
506	reports	62	92	0.143518519
507	request	53	146	0.122685185
508	requested	33	50	0.076388889
509	require	34	60	0.078703704
510	required	74	149	0.171296296
511	response	55	90	0.127314815
512	rest	37	58	0.085648148
513	results	54	156	0.125
514	return	73	56	0.168981481
515	rich	31	43	0.071759259
516	right	102	138	0.236111111
517	rights	28	16	0.064814815
518	risk	43	15	0.099537037
519	road	31	110	0.071759259
520	rom	30	20	0.069444444
521	run	36	54	0.083333333
522	s	274	1190	0.634259259
523	sales	72	11	0.166666667
524	same	95	288	0.219907407
525	save	84	13	0.194444444
526	saving	29	6	0.06712963
527	say	60	185	0.138888889
528	search	60	74	0.138888889
529	secret	27	13	0.0625
530	secrets	34	1	0.078703704
531	section	40	101	0.092592593
532	security	46	10	0.106481481
533	see	136	329	0.314814815
534	seen	44	83	0.101851852
535	select	33	45	0.076388889
536	selected	26	122	0.060185185

537	self	37	101	0.085648148
538	sell	56	2	0.12962963
539	selling	67	5	0.155092593
540	send	178	502	0.412037037
541	sending	55	46	0.127314815
542	sent	97	352	0.224537037
543	service	116	69	0.268518519
544	services	60	49	0.138888889
545	set	45	179	0.104166667
546	seven	40	24	0.092592593
547	several	61	233	0.141203704
548	sex	46	19	0.106481481
549	share	42	66	0.097222222
550	she	31	133	0.071759259
551	shipping	45	12	0.104166667
552	short	47	158	0.108796296
553	show	67	101	0.155092593
554	showing	26	38	0.060185185
555	shows	34	67	0.078703704
556	sign	31	64	0.071759259
557	signature	38	28	0.087962963
558	simple	85	94	0.196759259
559	simply	97	89	0.224537037
560	sincerely	38	26	0.087962963
561	site	118	213	0.273148148
562	sites	46	31	0.106481481
563	six	33	95	0.076388889
564	size	31	55	0.071759259
565	small	57	113	0.131944444
566	so	176	546	0.407407407
567	software	81	146	0.1875
568	sold	45	8	0.104166667
569	someone	62	102	0.143518519
570	something	43	134	0.099537037
571	soon	62	124	0.143518519
572	sound	27	116	0.0625
573	sources	37	83	0.085648148
574	special	91	308	0.210648148
575	start	105	61	0.243055556
576	started	64	40	0.148148148
577	starting	30	62	0.069444444
578	state	113	322	0.261574074
579	states	39	75	0.090277778
580	step	51	31	0.118055556
581	steps	27	13	0.0625
582	still	56	199	0.12962963
583	stop	39	51	0.090277778
584	street	38	93	0.087962963
585	subject	115	272	0.266203704
586	substantial	26	45	0.060185185

587	succeed	30	5	0.069444444
588	success	76	27	0.175925926
589	successful	43	77	0.099537037
590	suite	40	31	0.092592593
591	super	30	8	0.069444444
592	support	47	139	0.108796296
593	sure	77	112	0.178240741
594	system	65	270	0.150462963
595	t	191	529	0.44212963
596	take	142	287	0.328703704
597	takes	34	59	0.078703704
598	taking	32	71	0.074074074
599	talking	32	54	0.074074074
600	tax	30	7	0.069444444
601	technology	36	168	0.083333333
602	telephone	43	163	0.099537037
603	tell	76	89	0.175925926
604	tested	27	11	0.0625
605	than	125	535	0.289351852
606	thank	95	172	0.219907407
607	that	262	1215	0.606481481
608	their	141	633	0.326388889
609	them	146	323	0.337962963
610	themselves	28	79	0.064814815
611	then	145	263	0.335648148
612	there	172	783	0.398148148
613	these	152	589	0.351851852
614	they	159	519	0.368055556
615	thing	48	73	0.111111111
616	things	42	111	0.097222222
617	think	78	216	0.180555556
618	third	30	110	0.069444444
619	this	329	1317	0.761574074
620	those	93	385	0.215277778
621	thought	51	93	0.118055556
622	thousands	97	8	0.224537037
623	through	104	280	0.240740741
624	throughout	27	55	0.0625
625	time	212	446	0.490740741
626	times	59	84	0.136574074
627	tips	34	4	0.078703704
628	to	389	1948	0.900462963
629	today	142	79	0.328703704
630	told	29	57	0.06712963
631	toll	46	2	0.106481481
632	too	56	141	0.12962963
633	took	33	55	0.076388889
634	top	57	58	0.131944444
635	total	57	64	0.131944444
636	totally	36	16	0.083333333

637	track	29	49	0.06712963
638	travel	32	85	0.074074074
639	treat	29	12	0.06712963
640	trial	26	13	0.060185185
641	truly	36	23	0.083333333
642	try	74	72	0.171296296
643	trying	35	53	0.081018519
644	turn	41	43	0.094907407
645	type	100	166	0.231481481
646	u	62	307	0.143518519
647	under	60	185	0.138888889
648	understand	45	82	0.104166667
649	unique	36	40	0.083333333
650	united	38	73	0.087962963
651	unlimited	43	3	0.099537037
652	until	57	108	0.131944444
653	up	178	372	0.412037037
654	upon	32	91	0.074074074
655	us	190	467	0.439814815
656	use	139	503	0.321759259
657	used	67	336	0.155092593
658	using	97	265	0.224537037
659	value	29	48	0.06712963
660	ve	93	130	0.215277778
661	very	136	371	0.314814815
662	via	56	219	0.12962963
663	video	44	34	0.101851852
664	visa	51	31	0.118055556
665	visit	80	114	0.185185185
666	wait	36	11	0.083333333
667	waiting	41	6	0.094907407
668	want	165	147	0.381944444
669	wanted	33	37	0.076388889
670	was	107	440	0.247685185
671	watch	37	6	0.085648148
672	way	123	279	0.284722222
673	we	248	712	0.574074074
674	web	106	347	0.24537037
675	website	45	96	0.104166667
676	week	98	86	0.226851852
677	weekly	27	3	0.0625
678	weeks	64	99	0.148148148
679	well	108	482	0.25
680	were	75	278	0.173611111
681	what	178	545	0.412037037
682	when	131	305	0.303240741
683	where	107	358	0.247685185
684	while	60	229	0.138888889
685	who	139	609	0.321759259
686	whole	33	107	0.076388889

687	why	80	184	0.185185185
688	will	288	1091	0.666666667
689	win	36	4	0.083333333
690	windows	30	25	0.069444444
691	wish	85	121	0.196759259
692	with	294	1328	0.680555556
693	within	105	298	0.243055556
694	without	73	188	0.168981481
695	won	77	33	0.178240741
696	work	121	481	0.280092593
697	working	62	254	0.143518519
698	works	64	103	0.148148148
699	world	106	325	0.24537037
700	worldwide	36	14	0.083333333
701	worth	41	36	0.094907407
702	would	157	672	0.363425926
703	write	44	129	0.101851852
704	x	55	230	0.127314815
705	xxx	37	1	0.085648148
706	year	86	203	0.199074074
707	years	98	231	0.226851852
708	yes	65	57	0.150462963
709	you	391	786	0.905092593
710	your	332	450	0.768518519
711	yours	63	14	0.145833333
712	yourself	81	21	0.1875
713	zip	67	20	0.155092593