OSBF-Lua - A text classification module for Lua

The importance of the training method

Fidelis Assis

Abstract. OSBF-Lua is a C module for the Lua language which implements a Bayesian classifier enhanced with *Orthogonal Sparse Bigrams - OSB* - for feature extraction and *Exponential Differential Document Count - EDDC* – for feature selection. These two techniques, combined with the new training method introduced for TREC 2006 produce a highly accurate filter, yet very fast and economic in resources. OSBF-Lua is an Open Source Software available from http://osbf-lua.luaforge.net. *spamfilter.lua* is a production-class anti-spam filter available in the same package.

1 Introduction

The importance of feature extraction and feature selection is well known in token-based text classifiers. To address these points, OSBF-Lua uses the techniques *Orthogonal Sparse Bigrams (OSB)* [Siefkes et al. 2004] and *Exponential Differential Document Count (EDDC)* [Assis et al. 2005], respectively. The reading of the last reference is specially recommended for better understanding of this paper.

A third point, which is the subject of this paper, proved to deserve equal attention during the experiments for TREC 2006: *the training method*. The new training method introduced for TREC 2006, despite its simplicity, is the main factor responsible for the improvement in classification performance reached by OSBF-Lua, from the version presented in MIT Spam Conference 2006 to the present one submitted to TREC 2006 Spam Track.

2 Basic training methods

Statistic classifiers build their predicting models learning by examples. A basic training method is to start with an empty model, classify each new sample and train it in the right class if the classification is wrong. This is known as *Train On Error – TOE* [Yerazunis 2005]. An improvement to this method is to train also when the classification is right but the score is near the boundary, that is, *Train On or Near Error - TONE*. This method is also called *Thick Threshold Training* [Siefkes et al. 2004] [Yerazunis 2005].

The advantage of TONE over TOE is that it accelerates the learning process by exposing the filter to additional difficult (hard-to-classify) samples in the same period. Pure TONE was the training method used by OSBF-Lua before TREC 2006.

3 The new training method - *TONE with Header Reinforcement*

TONE with Header Reinforcement - TONE-HR – can be seen as an extension to TONE that adds a mechanism similar to white/black listing, in a sense that it uses information present in the header of the message for the hard-to-classify and hard-to-learn cases. Contrarily to normal white/black listing though, which is typically manual, *Header Reinforcement - HR* - is an entirely automatic process, from the detection of the cases where it applies to the selection of the most interesting features in the header to be considered.

HR extends TONE in the following way: after a message is trained as in TONE, the new score is calculated and the training is repeated, **now using only the header of the message**, while all three conditions below hold:

- the new score remains near the boundary;
- the absolute value of the variation of the score is less than a defined value;
- the number of repetitions is less than the maximum allowed.

The first condition is used to detect when HR applies, and then, together with the third and fourth, to avoid over training, which would result in bad score calibration. The limit values for these conditions were found experimentally and are documented in `spamfilter_commands.lua` source code, available in OSBF-Lua package. The code used for TREC evaluation is basically the same, but stripped down to the minimum necessary for the TREC runs.

The interesting aspect of this controlled repeated training, using only the header, is that instead of just two "colors", black and white, we get many more gradations between those extremes, producing better calibrated scores and, as a result, an improved area under the ROC curve. Another interesting characteristic is that it uses the normal training function already available in the filter and takes advantage of EDDC ability to automatically select, among the features present in the header, the most significant ones for classification.

Table 1 shows the evolution of OSBF from TREC 2005 [Assis et al. 2005] to the present version, and the improvement due to TONE-HR. The measurements were done against the TREC 2005 Full corpus.

Version	Training method	(1-ROCA)%
TREC 2005	TONE	0.019*
MIT Spam Conference 2006	TONE	0.016**
TREC 2006	**TONE-HR**	**0.010**

(*) Extra evaluation, by Prof. Gordon Cormack
(**) Better EDDC tuning

Table 1 – Evolution of OSBF-Lua

4 The packages submitted to TREC 2006

Four OSBF-Lua packages were submitted to TREC, differing only in the configuration of the thick threshold range and in the token delimiters used. The packages and their configurations are shown in Table 2:

Package	Thick threshold range	Token delimiter regex
oflS1F	[-15, 25]	`\s`
oflS2F	[-15, 25]	`[\s.@:/]`
oflS3F	[-20, 20]	`\s`
oflS4F	[-20, 20]	`[\s.@:/]`

Table 2 – Packages submitted to TREC 2006

The thick threshold ranges indicate the score region where reinforcement training is needed and the asymmetric range was set so that more good messages, proportionally to spam, were used for reinforcements. Scores in OSBF-Lua indicate "*hamminess*", rather than "*spamminess*" as expected by TREC scripts, that is, negative values indicate spam while non negative indicate ham. This explains the positive shift applied to the thick threshold range to reduce false positives. A conversion internal to the filter was implemented for compatibility with the values expected by TREC evaluation scripts.

The different combinations of thick threshold ranges and token delimiters were intended as an extra experiment to confirm the reduction in false positives induced by the asymmetric ranges in the table, without significant variation in global accuracy. As already observed during experiments, the results of TREC 2006 evaluation revealed the same behavior on both, public and private corpora, confirming that asymmetric ranges can be used for false positive reduction. This is useful since many users of spam

filters will consider the cost of misclassifying a ham message as spam higher than the cost of misclassifying a spam message.

All four packages were configured with two fixed-size databases, 46MB (4M buckets) each, one for spam and another for non-spam, totalizing 92MB. In practice we don't need such a large database, though. The default value in `spamfilter.lua` is only 1.1MB (92k buckets) per database, but the accuracy is still very good for practical purposes. Table 3 compares (1-ROCA)% values for small and large databases on the three public corpora: TREC 2005 Full, TREC 2006 English and TREC 2006 Chinese. Even with only 2.2 MB total database, the only significant change was on the full corpus, but still keeping a low value, comparable to the best in TREC 2005 Spam Track.

Database size	**(1-ROCA)%**		
	Full	**Public English**	**Public Chinese**
2.2 MB	0.022	0.058	0.003
92.0 MB	0.010	0.054	0.003

Table 3 – (1-ROCA)% for small and large databases

The submitted packages are available at http://osbf-lua.luaforge.net/TREC2006/, for easy reproduction of the results.

5 The ROC curve

The area under the ROC curve (AUC), or its complement (1-ROCA)%, is the main metric for ranking classifiers that has been used in TREC Spam Track. While it is a good measurement of the overall performance, it is not enough to assess classifiers when the ROC curves cross each other. For instance, low ham misclassification percentage (hm%) [Cormack, G. 2006], is more important than spam misclassification percentage (sm%) in spam filtering. hm% greater than 1%, to use a conservative value, is simply unacceptable. On the other hand, sm% greater than 10% is very poor for an anti-spam filter. So, the area restricted to the acceptable operation region, for instance where sm% < 10% and hm% < 1%, or even a more restricted one, considering the accuracy of present day spam filters, would be more appropriate when the ROC curves intersect.

Figure 1 shows ROC curves for the three versions listed in Table 1. TREC 2006 curve exhibits the best (1-ROCA)% value and is not intersected by any other, so it is clearly the best of the three classifiers. Because the other two curves intersect, the better (1-ROCA)% value of the version presented in MIT Spam Conference 2006 is not enough to tell if it is the best of the two. But a visual inspection shows that it dominates TREC 2005 version during most of the region where hm% is less than 1%, and confirms that it is the second best.

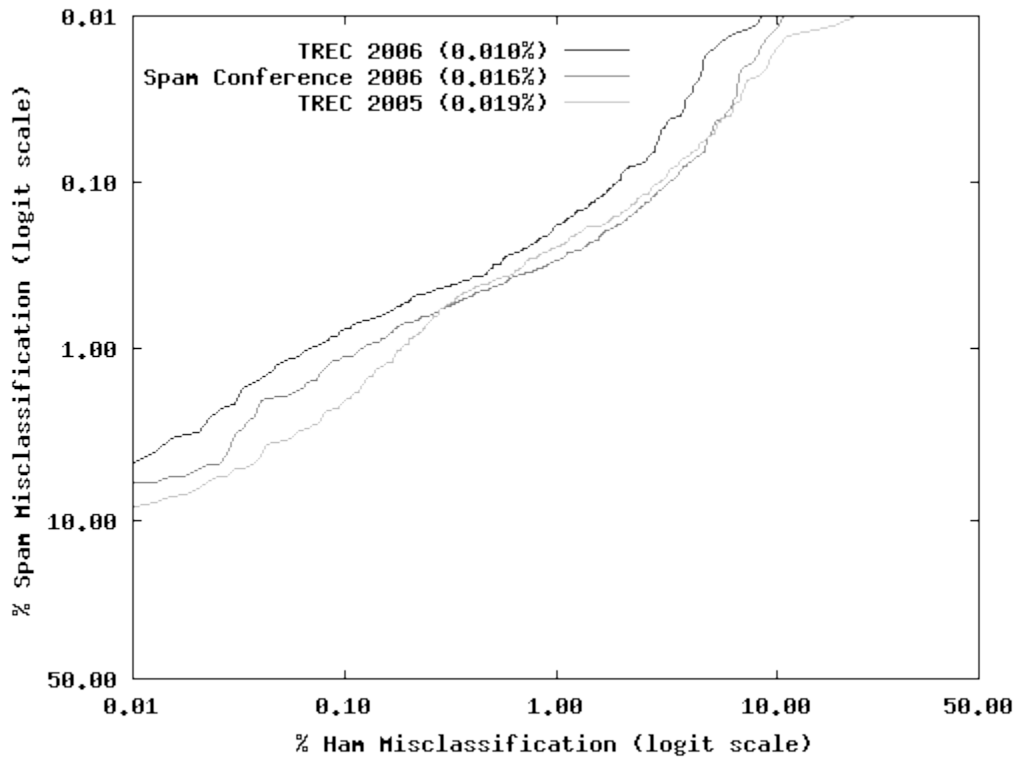

Figure 1 – ROC curves for the three versions in Table 1

6 TREC 2006 results

Table 4 shows the best filter for each of the top 5 teams, ranked by its (1-ROCA)% value on the immediate feedback run against the Aggregate pseudo-corpus [CORMACK, G. 2006]. OSBF-Lua (oflS1) was the best overall and also the best on each of the four corpora used for the tests, except on the Chinese one, where it was the second best.

Filter	**Aggregate**	**TREC06 English**	**TREC06 Chinese**	**MrX2**	**SB2**
oflS1	0.0295	0.0540	0.0035	0.0363	0.1300
tufS2	0.0370	0.0602	0.0031	0.0691	0.3379
ijsS1	0.0488	0.0605	0.0083	0.0809	0.1633
CRMS3	0.0978	0.1136	0.0105	0.1393	0.2983
hubS3	0.1674	0.1564	0.0353	0.2102	0.6225

Table 4 – TREC 2006 results - (1-ROCA)%

Since the tokenization done by present OSBF-Lua code is more suitable for occidental languages, its good result on the Chinese corpora was unexpected by the author. It depends on special delimiter characters (whitespace and possibly others) to split a text into tokens (words), but Chinese and other oriental languages do not generally use whitespace to separate tokens. This result can probably be credited to the new training method, because it makes special use of the header of the message, which is mostly independent of the language.

Table 5 shows the results of the four OSBF-Lua variants in the immediate feedback run. oflS1F and oflS3F were the first and second best with respect to (1-ROCA)%, but the variations are not statistically significant.

Corpora	oflS1F	oflS2F	oflS3F	oflS4F
MrX2	**0.0363 (0.0220-0.0597)**	0.0525 (0.0289-0.0956)	0.0523 (0.0300-0.0909)	0.0718 (0.0388-0.1327)
B2	**0.1300 (0.0752-0.2248)**	0.1479 (0.0920-0.2378)	**0.1249 (0.0715-0.2180)**	0.1407 (0.0735-0.2689)
Public English	**0.0540 (0.0343-0.0852)**	0.0597 (0.0357-0.0996)	0.0562 (0.0346-0.0911)	0.0583 (0.0387-0.0878)
Public Chinese	**0.0035 (0.0014-0.0085)**	0.0104 (0.0048-0.0226)	**0.0035 (0.0014-0.0089)**	0.0077 (0.0037-0.0163)

Table 5 – Results of the four OSBF-Lua variants - (1-ROCA)%

7 Conclusions

Training methods play an important role for the accuracy of adaptive anti-spam filters, side by side with techniques for feature extraction, feature selection and weighting for token-based filters, and deserve the same attention.

A training method for statistic anti-spam filters was introduced, *TONE-HR,* with experimental results that demonstrate its great contribution to the overall accuracy of OSBF-Lua. The author has no knowledge of a similar training method in the literature and believes that it is new, despite its simplicity.

Acknowledgements

I want to thank many people who have been contributing to the improvement of OSBF-Lua, by testing, packaging, mirroring and making useful suggestions and contributions. Among them, Alessandro Martins, Cassiano Aquino, Marcus Maciel, Pavel Kolar and Steve Pellegrin.

I want to thank the Luaforge team for hosting the OSBF-Lua project and André Carregal for his support with the questions related to this hosting.

I want also to thank William Yearazunis for creating the CRM114 project, which motivated me to dedicate time to anti-spam filters.

Finally, a special thanks to Christian Siefkes, for his time in reviewing this paper thoroughly and making many helpful and inspired comments and contributions to the text.

References

Assis, F., Yerazunis, W., Siefkes, C. and Chhabra S. 2005. CRM114 versus Mr. X: CRM114 Notes for the TREC 2005 Spam Track. In *The Fourteenth Text REtrieval Conference – TREC Spam Track 2005*. http://trec.nist.gov/pubs/trec14/papers/crm.spam.pdf.

Assis, F., Yerazunis, W., Siefkes, C. and Chhabra, S. 2006. Exponential Differential Document Count: A Feature Selection Factor for Improving Bayesian Filters Accuracy. In *2006 Spam Conference*, Cambridge, MA, 2006. http://osbf-lua.luaforge.net/papers/osbf-eddc.pdf.

Cormack, G. 2006. TREC 2006 Spam Track Overview. In NIST Special Publication: The Fifteenth Text REtrieval Conference Proceedings (TREC 2006). http://trec.nist.gov/pubs.html

Siefkes, C., Assis, F., Chhabra, S. and Yerazunis, W. 2004. Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering. In *European Conference on Machine Learning (ECML) / European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, September 2004. http://page.mi.fu-berlin.de/~siefkes/papers/winnow-spam.pdf.

Yerazunis, B. 2005. CRM114 Revealed – Or How I learned To Stop Worrying and Trust My Automatic Monitoring Systems ; this is the complete CRM114 manual available for free download at http://crm114.sourceforge.net.