BUPT at TREC 2006: Spam Track

Zhen Yang, Wei Xu, Bo Chen, Weiran Xu, and Jun Guo

PRIS Lab, School of Information Engineering,
Beijing University of Posts and Telecommunications, 100876, Beijing, China
yangzhen@pris.edu.cn

Abstract. This report summarizes our participation in the TREC 2006 spam track, in which we consider the use of Bayesian models for the spam filtering task. Firstly, our anti-spam filter, Kidult, is briefly introduced. And then we try to use weighted adjustment of separating hyperplane and selective classifiers ensemble to improve the filtering performance. Finally, we summarize the relevant results from the official evaluation.

1 Introduction

In 2005, a new track on spam filtering was introduced to TREC, whose goal was to provide a standard evaluation of current and proposed spam filtering approaches. "The 2006 track reprises the 2005 experiments with new filters and data, and also investigate delayed feedback and active learning. [1]" There are two tasks:
1. Online filtering - enhancement to TREC 2005 task;
2. Active learning - completely new task.

In this year, we focus on the online filtering task. For the pilot task, active learning, the only difference is that the "run.sh" is replaced by the active learning shell "active.cpp" with random selection in the jig. So the remainder of this paper is structured around online filtering task. Section 2 outlines an briefly overview of the "kidult" anti-spam framework. Some improvements are proposed in Section 3. In Section 4, we summarize the relevant results from the official evaluation. The major conclusions that can be drawn from the evaluation are presented in Section 5.

2 System Overview

The "kidult" is an anti-spam solution with self-dependence intellectual property, which is developed by Pris Lab of Beijing University of Posts and Telecommunications. The resulting technology of "kidult" has been successfully released in our TREC 2005 and TREC 2006 spam track system [2].

The processing procedure of the "kidult" system is same as the general processing framework of TREC 2006 spam track. Our system uses Bayesian models for email classification. The Bayesian classifier is a probability based approach, which is often used in text classification applications and experiments for its simplicity and effectiveness. The following subsections describe our methods in greater detail.

2.1 Preprocessing

Some common or often proposed initial transformations are: lookalike transformations, HTML deobfuscation, MIME normalization, character set folding, case folding, word stemming, stop words list, feature selection [3]. Discussed in our 2005 spam track report [2] and CRM114's notes [4], it would be far better if the learning machine itself either made these transformations automatically or used all the features. In this literature, in this work, we only use HTML deobfuscation and MIME normalization.

2.2 Chinese Word Segmentation

Usually, the basic unit for text processing is word. It is natural for English, but for Chinese language text, words are not demarcated in a sentence. Thus, word segmentation must be performed first in most natural language processing (NLP) applications, which is necessary but time-consuming. We adopted a POC-NLW based HMM segmenter, as described in [5], to implement the preprocessing of the context of an email. However, in order to meet the constraints on processing time, only a simplest segmentation model was used, which was a purely character-level tagger based on the POC-NLW template without any word-level information. This model only need to load fewest features and the loading can be accomplished in far less than one second, while other more complex models cost a few seconds on feature loading. However, this simplification may lead to decay on the overall performance. As presented in [5], detailed experimental results show that such a simplified model performs much worse than those complex ones.

2.3 Tokenization

Usually, the word is used as the basic processing unit. The basic idea is to break of the input text stream into a series of tokens. The boost [6] Tokenizer package provides a flexible and easy to use way to break of a string or other character sequence into a series of tokens, by which we can choose how the string gets broken up using different Tokenizer function. In this work, we break up the input text string based on a superset of comma separated value lines (such as space, punctuation, customize escaped list separator and offset separator).

2.4 Naive Bayes Spam Filtering Framework

The Bayesian classifier is a probability based approach, which is often applied to text categorizations tasks. For spam detection, suppose each email instance M is described by a conjunction of word attribute values $< w_1, w_2, ..., w_n >$. And L is the number of target classes ($C_i, i = 1, ..., L$). The basic concept of Bayesian classifier is to find whether an e-mail is spam or not by looking at which words are found and which words are absent from the message. In the literature, the Bayesian approach to the new email is to assign the most probable target label:

$$\begin{aligned} H_{MAP} &= \arg\max_{i \in L} P(C_i \mid w_1, w_2, ..., w_n) \\ &= \arg\max_{i \in L} P(C_i) P(w_1, w_2, ..., w_n \mid C_i) \end{aligned} \quad \textbf{(1)}$$

To makes the estimation of parameters tractable, the Naive Bayes assumption is used, which suppose that the attribute values are conditionally independently, then

$$H_{NB} = \arg\max_{i \in L} P(C_i) \prod_k P(w_k \mid C_i) . \quad \textbf{(2)}$$

For the situation of spam detection, attribute values $< w_1, w_2, ..., w_n >$ is the words in one email message (for Chinese corpus, word segmentation is needed), where L is the number of target classes C_i (e.g. C_+ spam/C_- ham). In practice， log-likelihood is computed as following:

$$\text{score(M)} = \log P(C_+) + \sum_k \log P(w_k \mid C_+) - (\log P(C_-) + \sum_k \log P(w_k \mid C_-)) \,. \tag{3}$$

Therefore, if score(M) > 0, the email will be assigned to C+, and C- otherwise. In our experiments, n-gram model shows good performance. But with the increase of n, n-gram suffered from data sparseness and real-time limitation, which makes higher order model cannot be used in our submitted systems.

2.5 Add-One Smoothing Algorithm and Kill-One Strategy

The statistical approaches for spam filtering are often Bayesian and several distribution models (such as multi-variants Bernoulli model, Poisson Naive Bayes model, and the multinomial model) are assumed. The difference between these models is the ways of calculating $P(w_k|C_i)$. In this work, multinomial model is used for its superior performance [7].

One benefit of the multinomial approach is the number of available smoothing methods to handle unhappened tokens. In Bayesian models, according to the principles of symmetry, the tokens have no other characteristics in addition to the number of token. Then token k with the same counter has the same probability value. Suppose n_r is the number of special token occurred as often as *r* in training corpus. N is the total number of tokens, then:

$$\sum_r r \cdot n_r = N \tag{3}$$

Based on the Maximum Likelihood (ML) estimation model, the number of w_k in training corpus is $N(w_k) = r$. Then $P^r_{ML}(w_k) = r / N$, subject to:

$$\sum_r n_r P^r_{ML}(w_k) = 1 \tag{4}$$

For simplicity we use add-one formula for smoothing [8], which use the r=1 to estimate the unhappened token:

$$P^0_{ML} = P^1_{ML} = 1/N \tag{5}$$

On one hand, for our preprocessing strategy, many insignificant and meaningless tokens are often produced, which increase the system load. By using the add-one smoothing algorithm, we can discard the tokens with r=1, which doesn’t decreasing the filtering performance. It is so-called kill-one strategy. In practice, the tokens with r=1~3 are usually discarded. On the other hand, tokens’ discarding is triggered by setting conditions (such as run time limitation, memory). The effection on precision of our system still needs to be observed.

3 Improvements

3.1 Weighted Adjustment of Separating Hyperplane

In our 2005 spam track, we discussed some improvements based on separating hayperplane weighted adjustment [2]. The official evaluation results of TREC 2005 show that the modification is effective. So in the 2006 track, we reprise the 2005 methods with new tasks and data.

3.2 Selective Classifiers Ensemble

Last year, we discuss the Bagging-based method for spam filtering. In this year, we use selective ensemble to improve the performance of classifying. After analysis of the relationship between the ensemble and its component, some researchers [9,10,11] reveal that it may be better to ensemble many instead of all of the classifiers at hand. Selective classifiers ensemble is thought an improved method for Bagging aggregate, in which mutual information weighted method is widely used [9,10,11]. For this year's track, we discuss two aggregate strategies: 1) selective ensemble based on mutual information of each classifier; 2) selective ensemble based on mutual information sharing with the optimal classifier.

4 Experiments

In this section, we report the relevant results from the official evaluation. The basic statistics for these datasets are given as following: MrX2 (9032 ham, 40135 spam), SB2 (9274 ham, 2751 spam). The performance of "kidult" anti-spam solution is given in Table 1-Table 2. Results are included for 2 corpora, with immediate feedback, delayed feedback, and active learning as denoted by the run tag suffix: x2 (MrX2 corpus, immediate feedback), x2d (MrX2 corpus, delayed feedback), x2a (MrX2 corpus, active learning), b2 (SB2 corpus, immediate feedback), b2d (SB2 corpus, delayed feedback), b2a (SB2 corpus, active learning).

Table 1. Immediate/delay feedback results

Run tag	Ham Misc%	Spam Misc%	Lam%	(1-ROCA)%
KB3S1x2	9.90 (9.29-10.54)	0.68 (0.60-0.76)	2.66 (2.47 - 2.86)	2.5926 (2.3609 - 2.8465)
BASS2x2	10.62 (9.99-11.27)	0.56 (0.49-0.64)	2.52 (2.35 - 2.71)	2.5486 (2.3071 - 2.8147)
B53S3x2	9.49 (8.90-10.12)	0.65 (0.57-0.73)	2.55 (2.38 - 2.73)	2.3501 (2.1435 - 2.5762)
KB9S4x2	10.32 (9.70-10.97)	0.58 (0.51-0.66)	2.53 (2.37 - 2.71)	2.5100 (2.2949 - 2.7446)
KB3S1x2d	13.76 (13.06-14.49)	0.71 (0.63-0.80)	3.27 (3.09 - 3.46)	3.6977 (3.4081 - 4.0109)
BASS2x2d	11.51 (10.85-12.18)	0.74 (0.65-0.82)	3.01 (2.80 - 3.24)	2.9571 (2.7133 - 3.2221)
B53S3x2d	9.13 (8.54-9.74)	1.90 (1.77-2.04)	4.22 (4.05 - 4.41)	3.0866 (2.8526 - 3.3391)
KB9S4x2d	13.42 (12.72-14.14)	0.68 (0.60-0.76)	3.15 (2.96 - 3.35)	3.4217 (3.1687 - 3.6942)
KB3S1b2	2.30 (2.00-2.62)	3.27 (2.64-4.01)	2.74 (2.39 - 3.15)	1.5545 (1.2901 - 1.8720)
BASS2b2	2.10 (1.82-2.42)	3.16 (2.54-3.89)	2.58 (2.30 - 2.90)	1.4311 (1.1936 - 1.7151)

B53S3b2	2.61 (2.29-2.95)	3.56 (2.90-4.32)	3.05 (2.72 - 3.41)	1.6350 (1.3608 - 1.9634)
KB9S4b2	2.66 (2.35-3.01)	3.02 (2.41-3.73)	2.83 (2.48 - 3.24)	1.4970 (1.2363 - 1.8117)
KB3S1b2d	3.69 (3.31-4.09)	5.45 (4.63-6.37)	4.49 (4.06 - 4.96)	2.9271 (2.5474 - 3.3613)
BASS2b2d	3.64 (3.27-4.05)	5.45 (4.63-6.37)	4.46 (4.04 - 4.93)	2.9050 (2.5229 - 3.3430)
B53S3b2d	3.86 (3.48-4.27)	5.63 (4.80-6.56)	4.67 (4.29 - 5.07)	3.0487 (2.6378 - 3.5213)
KB9S4b2d	4.83 (4.40-5.29)	4.54 (3.80-5.39)	4.69 (4.24 - 5.17)	3.0337 (2.6993 - 3.4081)

Table 2. Active learning results

Run tag	Ham Misc%	Spam Misc%	Lam%	(1-ROCA)%
KB3A1x2 Teach=100	8.41 (6.52-10.63)	21.68 (20.44-22.97)	13.75 (12.06 - 15.65)	10.0451 (8.644 - 11.644)
KB3A1x2 Teach=25600	4.67 (3.28-6.44)	1.20 (0.89-1.58)	2.38 (1.86 - 3.04)	1.1716 (0.6888 - 1.9860)
BASA2x2 Teach=100	8.95 (7.00-11.22)	19.38 (18.19-20.61)	13.32 (11.59 - 15.27)	8.8997 (7.4168 - 10.645)
BASA2x2 Teach=25600	4.27 (2.94-5.98)	1.15 (0.85-1.52)	2.23 (1.81 - 2.74)	1.1815 (0.7694 - 1.8104)
KB9A3x2 Teach=100	9.21 (7.24-11.51)	20.56 (19.34-21.82)	13.94 (12.29 - 15.78)	9.0909 (7.628 - 10.801)
KB9A3x2 Teach=25600	2.40 (1.43-3.77)	2.09 (1.67-2.57)	2.24 (1.73 - 2.90)	0.9953 (0.5654 - 1.7463)
WEIA4x2 Teach=100	9.75 (7.72-12.10)	21.11 (19.88-22.38)	14.53 (13.16 - 16.02)	9.5714 (8.304 - 11.0094)
WEIA4x2 Teach=25600	3.07 (1.96-4.57)	1.58 (1.23-2.01)	2.21 (1.74 - 2.79)	1.0979 (0.6643 - 1.8093)
KB3A1b2 Teach=100	10.16 (8.07-12.59)	14.95 (11.86-18.48)	12.36 (10.56 - 14.41)	9.8034 (7.860 - 12.164)
KB3A1b2 Teach=6400	2.75 (1.69-4.21)	1.05 (0.34-2.44)	1.70 (1.03 - 2.80)	1.3942 (0.8545 - 2.2668)
BASA2b2 Teach=100	22.12 (19.15-25.31)	10.74 (8.10-13.87)	15.60 (13.51 - 17.94)	12.3143 (10.297 - 14.662)
BASA2b2 Teach=6400	2.61 (1.58-4.05)	1.26 (0.46-2.73)	1.82 (1.07 - 3.07)	1.4156 (0.7990 - 2.4960)
KB9A3b2 Teach=100	32.55 (29.16-36.09)	5.89 (3.95-8.41)	14.81 (12.26 - 17.79)	13.2157 (11.429 - 15.234)
KB9A3b2 Teach=6400	3.98 (2.68-5.67)	3.37 (1.94-5.41)	3.66 (2.73 - 4.91)	1.4717 (0.8074 - 2.6676)
WEIA4b2 Teach=100	19.23 (16.43-22.28)	10.74 (8.10-13.87)	14.47 (12.65 - 16.51)	13.0602 (11.166 - 15.221)
WEIA4b2 Teach=6400	2.88 (1.79-4.38)	1.05 (0.34-2.44)	1.75 (nan - nan)	1.4083 (0.8118 - 2.4323)

5 Summary

For the run time limitation of spam track, filters that use more than 2 seconds per message will be killed and the result will be recorded as "class=ham score=0" for any unprocessed messages. This makes us use simplified algorithms. In experiments, some methods with good performance but time-consuming can not be applied. More importantly, the improvement of our system more and more depends on the details, such as word segmentation, HTML deobfuscation, MIME normalization, character set folding, etc., which already have departure from the original goal of TREC in some degree.

6 Acknowledgements

This research is partially supported by NSFC (National Natural Science Foundation of China) under Grant No.60475007 and No.60675001, Key Project of Chinese Ministry of Education under Grant No.02029 and the Foundation of Chinese Ministry of Education for Century Spanning Talent.

7 References

1. http://plg.uwaterloo.ca/~gvcormac/spam/
2. Yang, Z., Xu, W.R., Chen, B., Hu, J.N., Guo, J.: PRIS Kidult Anti-SPAM Solution at the TREC 2005 Spam Track: Improving the Performance of Naive Bayes for Spam Detection. Proceedings of Fourteenth Text REtrieval Conference (2005)
3. Yerazunis, W., Chhabra, S., Siefkes, C., Assis, F., Gunopulos, D.: A Unified Model of Spam Filtration. http://crm114.sourceforge.net/UnifiedFilters.pdf
4. Assis, F., Yerazunis, W., Siefkes, C., Chhabra, S.:CRM114 versus Mr. X: CRM114 Notes for the TREC 2005 Spam Trac. Proceedings of Fourteenth Text REtrieval Conference (2005)
5. Chen, B., Peng, T., Xu, W.R., Guo, J.: POC-NLW Template for Chinese Word Segmentation. Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing (2006) 177–180
6. http://www.boost.org/
7. Kim, Y.H., Hahn, S.Y., Zhang, B.T.: Text Filtering by Boosting Naive Bayes Classifiers. In SIGIR Conference on Research and Development (2000)
8. Nicolas, G., Domingo, O.: Improving Multiclass Pattern Recognition by the Combination of Two Strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28(2006) 1001–1006
9. Zhou, Z.-H., Wu, J.-X., Tang, W.: Ensembling Neural Networks: Many Could Be Better Than All. Artificial Intelligence (2002)239–263
10. Zhou, Z.-H., Wu, J.-X., Tang, W., Chen Z.-Q.: Selectively Ensembling Neural Classifiers. Proceedings of the International Joint Conference on Neural Networks (2002) 1411–1415
11. Strehl, A., Ghosh, J.: Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. Journal on Machine Learning Research (2002) 583–617