Joint NLP Lab between HIT^2 at CEAS Spam-filter Challenge 2008

Haoliang Qi
Heilongjiang Institute of Technology
No. 999, Hongqi Street,
Harbin,P.R. China, 150050,
+86-451-88627961
Haoliang.qi@gmail.com

Xiaoning He
Harbin University of Science and Technology
No. 52 Xuefu Road,
Harbin, P.R.China, 150080
+86-451-86390114
nxnh@qq.com

Muyun Yang
Harbin Institute of Technology
No.92, West Da-Zhi Street,
Harbin, P.R.China, 150001
+86-451-86412449
ymy@mtlab.hit.edu.cn

Jun Li
Heilongjiang Institute of Technology
No. 999, Hongqi Street,
Harbin,P.R. China, 150050,
+86-451-88627961
leejunemail@163.com

Guohua Lei
Heilongjiang Institute of Technology
No. 999, Hongqi Street,
Harbin,P.R. China, 150050,
+86-451-88628518
islgh@126.com

Sheng Li
Harbin Institute of Technology
No.92, West Da-Zhi Street,
Harbin, P.R.China, 150001
+86-451-86412449
lisheng@hit.edu.cn

ABSTRACT

This paper reports our participation of CEAS Spam-filter Challenge 2008. The logistic regression model, n-gram and TONE (Train On /Near Error) were used to build the systems. We improved the weighting method which reduces the impact of the features appearing both in spam messages and ham messages. . We achieved competitive results in all tasks and got the first in a subtask of Lab Evaluation Task.

1. INTRODUCTION

This is the first year that the group participating Conference on Email and Anti-Spam (CEAS) Spam-filter Challenge 2008, and we took part in the CEAS Spam-Filter Challenge Live Spam Task and the CEAS Spam-Filter Challenge Lab Evaluation Task. The most members of the group are from Joint NLP (Natural Language) Lab between HIT^2 (Harbin Institute of Technology and Heilongjiang Institute of Technology) except Xiaoning He, who is a master student in Harbin University of Science and Technology.

The logistic regression model, n-gram and TONE (Train On /Near Error) were used to build the systems. We achieved competitive results in all tasks and got the first and the second in the 108.1.short task which is one of Lab Evaluation Task.

2. SYSTEM DESCRIPTION

One system was used to online Live Task and 2 systems were used to Lab Evaluation Task. We use HITLR to denote the system used for Live Task and Hao1 and Hao2 for the systems of Lab Evaluation Task. The filtering part of HITLR is same to the Hao2 system for Lab Evaluation Task. The main difference is Exim4, the default MTA (Message Transfer Agent) in Debian Linux Operating System. Exim4 is used to deal with messages.

When building a spam filter, there are 3 problems: email presentation (i.e. feature extraction), filtering model and training method. The followings will present our solutions for these problems.

CEAS 2008 –Fifth Conference on Email and Anti-Spam, August 21-28, 2008, Mountain View, California USA

2.1 Feature Extraction

When Extracting features from email, overlapping character-level *n*-grams is used [1]. For example, for a string "abcd", the bigrams of this string are "ab", "bc" and "cd". In the competition, 4-gram was used for all of our systems. Furthermore, with email data, we reduce the impact of long messages by considering only the first 3,000 characters of each message [1]. No other feature selection or domain knowledge was used. For a certain n-gram, if it appears in the message, its value is 1, otherwise 0.

2.2 Filtering Model

Filtering models can roughly be divided into two types: generative models (like Naive Bayes), and discriminative models (like Support Vector Machines and Logistic Regression (LR).) In most text classification tasks, discriminative models have outperformed generative models. We followed Ref. [2][3], LR is used as the filtering model. So we can predict a message by following Equation 1.

$$P(\mathrm{Y}=\mathrm{spam}\mid\vec{\mathrm{f}})=\frac{e^{\sum w_i f_i}}{1+e^{\sum w_i f_i}} \qquad (1)$$

Where $\vec{\mathrm{f}}=\{f_1, f_2,\ldots, f_n\}$ is the message's features, w_i is its weight.

2.3 Training Method

When training the spam filter, we use TONE method [4][5]. This method is also called Thick Threshold Training. Training instances are re-trained even if the classification is correct with a score near the threshold θ. In this way, a large margin classifier will be trained that is more robust when classifying borderline instances.

We improved the LR algorithm according to the characteristic of spam filtering. The improved methods reduce the impact of the features appearing both in spam messages and ham messages. We will present two methods to achieve the goal; one adjusts update weight, the other directly reduces the feature’s weight.

2.3.1 Adjusting Update Weight

Given a feature f_i, the ratio of its weight to be adjusted is

$$weight\_adj\_ratio = 1 - \mathrm{abs}\left(\frac{p(\mathrm{spam})-p(\mathrm{ham})}{p(\mathrm{spam})+p(\mathrm{ham})}\right)^{n} \quad (2)$$

where p(spam) is the probability of feature f_i in spam messages, and p(ham) is the probability of feature f_i in ham messages. abs(x) computes the absolute value of a specified number x. n is set to 2 in the experiments.

According to LR model, the adjusted feature’s weight is computed as

$$\begin{aligned}&\text{if (SPAM)}\\&\qquad weight\_adj = weight\_adj\_ratio \;*\; (1 - p) \;*\; \mathrm{RATE};\\&\text{else}\\&\qquad weight\_adj = weight\_adj\_ratio \;*\; p \;*\; \mathrm{RATE};\end{aligned} \quad (3)$$

where the RATE is learning rate in LR model.

Then the feature’s final weight is

$$weight = \begin{cases} 0 & \mathrm{abs}(weight\_adj) > \mathrm{abs}(weight) \\ \mathrm{abs}(ori\_weight) - \mathrm{abs}(weight\_adj) & \text{otherwise} \end{cases} \quad (4)$$

where the original weight can be computed by LR model.

This improved algorithm was used in HITLR system and Hao2 system.

2.3.2 Directly reducing the Feature’s Weight.

Now we present the second improvement which directly reduces the feature’s weight.

The adjust ratio of the feature f_i is defined as

$$weight\_adj\_ratio = \mathrm{abs}\left(\frac{p(\mathrm{spam})-p(\mathrm{ham})}{p(\mathrm{spam})+p(\mathrm{ham})}\right) \quad (5)$$

Then the feature’s final weight is

$$weight = ori\_weight \;*\; weight\_adj\_ratio \quad (6)$$

where the original weight can be computed by LR model.

This improved algorithm was used in Hao1 system.

3. EXPERIMENTS AND RESULTS

No external resource is used in the competition. The initial weights of all the systems are set to 0. Table 1 shows some statistics of the test corpus.

Table 1. Test Corpus

Task	All Messages	Spam Messages	Ham Messages
CEAS 2008	137704	110579	27125
CEAS 2008 (Short)	127925	103262	24663

CEAS 2008 (Short) is referred to a special truncated version of CEAS 2008, which terminates before an outbreak of the CNN virus caused several incorrect feedback responses.

Only one system, i.e. HITLR, is used to take part in Live Spam Task. We submitted 2 systems (Hao1 and Hao2) to take part in Lab Evaluation.

There are 3 tasks in Lab Evaluation.

1. A replay of the messages used in the CEAS Live Spam Task, in the same order, including feedback. The result for this task is labeled as l08.1.

2. An "active learning" task in which the filter receives immediate feedback for 1000 messages of its own choosing. There are 2 subtasks, which are "a1000" and "b1000". The difference between the "a1000" and "b1000" is that the "a1000" files are scored on all messages whereas the "b1000" files exempt the 1000 messages for which the filter requests a label.

3. Tasks 1 and 2 are repeated on a different, private dataset that may be more realistic than the CEAS live stream. And the results is not released until now.

Table 2 shows our results. Hao2.l08.1.short and Hao1.l08.1.short got the first and the second on short corpus in Task 1 of Lab Evaluation.

For active learning tasks, the improved methods may have side effect. The performance of our systems is lower than the other LR systems.

Comparing the results between HITLR and Hao2.l08.1, we can see that the timeout has fatal effect on the performance, because the filters lose the learning opportunity.

Table 2. Competition Results

RunID	Timeout	LAM(%)	1-ROCA(%)
HITLR	0.00097	0.389	0.0403
Hao1.l08.1	--	0.31	0.0197
Hao2.l08.1	--	0.26	0.0277
Hao1.l08.1.short	--	0.15	0.0050
Hao2.l08.1.short	--	0.12	0.0046
Hao1.a1000.1	--	0.51	0.0557
Hao2.a1000.1	--	0.43	0.0303
Hao1.a1000.1.short	--	0.17	0.0102
Hao2.a1000.1.short		0.13	0.0039
Hao1.b1000.1	--	0.43	0.0459
Hao2.b1000.1	--	0.37	0.0226
Hao1.b1000.1.short	--	0.19	0.0127
Hao2.b1000.1.short	--	0.15	0.0045

4. REFERENCES

[1] D. Sculley and G. M. Wachman. *Relaxed Online SVMs for Spam Filtering*. SIGIR’07

[2] J. Goodman. *Online Discriminative Spam Filter Training*. CEAS2006.

[3] G. V. Cormack. *University of Waterloo Participation in the TREC 2007: Spam Track*. TREC 2007.

[4] Sieekes C., Assis F., Chhabra S. *et al. Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering*. European Conference on Machine Learning

(ECML) /European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD).September 2004.

[5] Fidelis Assis. *OSBF-Lua - A Text Classification Module for Lua The Importance of the Training Method*. TREC 2006. 2006.