Attacking Statistical Spam Filters

Greg Wittel

&

Felix Wu

Outline

- *Introduction*
- Background
- Attacking Filters
- Testing a New Attack
- Conclusion

The Problem

- Spam -- unsolicited bulk e-mail
- Spam volume increasing:
  - About 64% of e-mail is spam (~48% a year ago) [Brightmail]
- Costs: Equipment, bandwidth, time.
- Dubious (and inappropriate) content
- Effects utility of e-mail

Possible Solutions

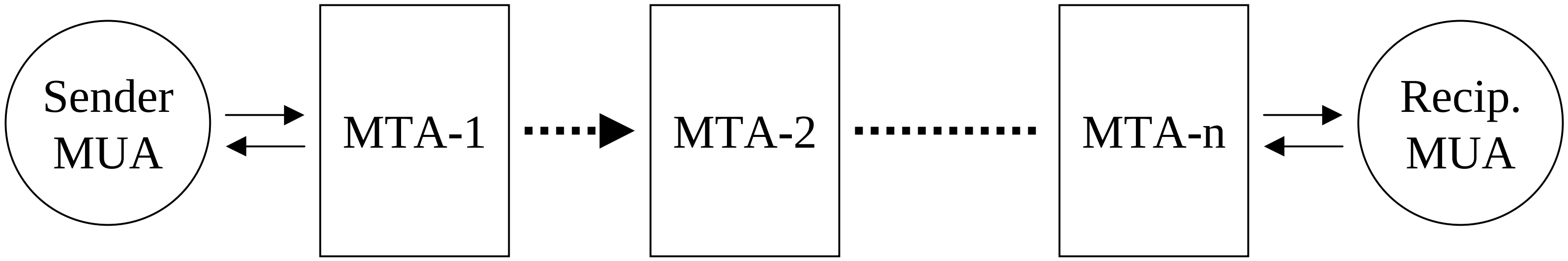
- Legal
  - Hard to enforce
  - Will not stop spammers without enforcement
- Technical
  - New protocols - authentication, authorization.
  - E-postage (monetary or CPU time)
  - Filtering

Outline

- Introduction
- *<u>Background</u>*
- Attacking Filters
- Testing a New Attack
- Conclusion

Mail System in Brief

- SMTP used between MTAs
- POP3/IMAP used for MUA->MTA Comms.
- Spam may originate at an ‘MTA’
  - e.g. Compromised or trojaned machines



Why Filtering?

- Can’t control outside networks.
- “Easy” short term fix against incoming traffic.
- SMTP upgrade a (hard) long term issue.
- Can hide junk from end users and cut some downstream costs.

Problems with Filtering

- Becoming an arms race
- Filtering overhead
- Does not eliminate cost of spam
- Not necessarily easy to implement

Filtering Overview

- Roots in text classification
- General filtering methods:
  - Rule based
  - Statistical
  - Hybrid

Rule Based Filters

- Classifies a message based on whether or not it meets a series of criteria.
- Hand made or algorithmically generated
- Simple Rules:
  - Checksums, white/black/greylists, keywords
- Complex rules:
  - e.g. “Does this message have lots of HTML comments?”

Statistical Filters

- Driven by statistics derived from data
- Attempts to find a statistical difference between different message classes.
  - e.g. word frequency, document length
- Often machine learning driven
  - Naïve bayesian, k-NN, SVM

Document Representation

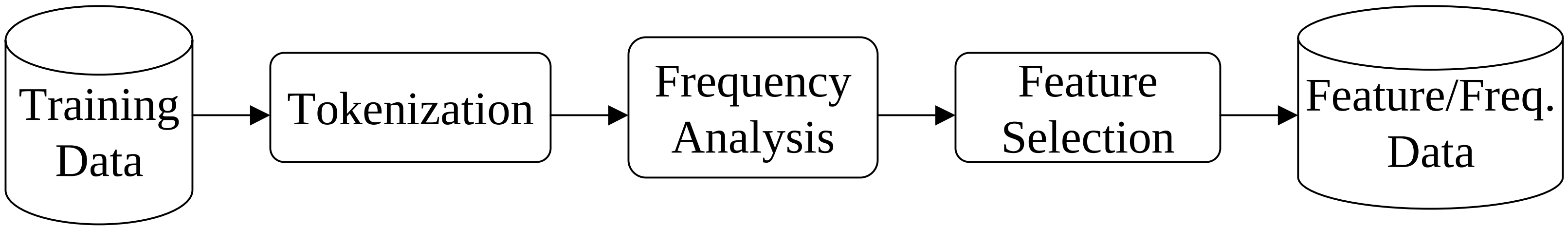
- Statistical filters often use ‘bag of words’ model
- Documents reduced to a numeric feature vector
- Features may represent a word, phrase, or information derived from the document.

Document Representation

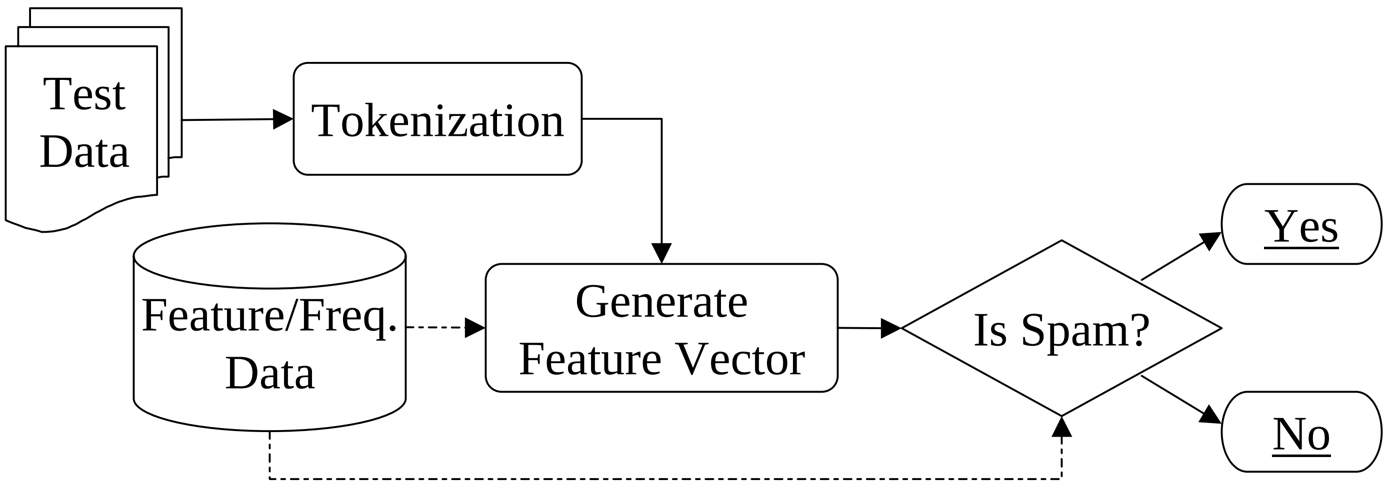
- Documents classified based on feature vector
- Issues:
  – Vector can lose relationship between features
  – May have tens of thousands of features

Classification Model

- Training Stage



- Testing



Outline

- Introduction
- Background
- *Attacking Filters*
- Testing a New Attack
- Conclusion

Attack Classes

- Attempted attack methods:
  - Tokenization
    - Works against feature selection by splitting or modifying key message features
    - e.g. Splitting up words with spaces, HTML tricks
  - Obfuscation
    - Use encoding or misdirection to hide contents from filter
    - e.g. HTML/URL encoding, letter substitution

Attack Classes cont.

– Weak Statistical
  - Skew message statistics by adding in random data
  - e.g. Add in random words, fake HTML tags, random text excerpts

– Strong Statistical
  - Differentiated from ‘weak’ attacks by using more intelligence in the attack
  - Guessing v. educated guessing
  - e.g. Graham-Cumming Attack

Attack Classes cont.

– Misc:
  - Sparse Data attack
  - Hash breaking attacks

Graham-Cumming Attack

- Regular dictionary attack didn’t work so he:
  - Added random dictionary words to a spam
  - Tested variations against filter
  - Found common set of words that were able to weaken ‘spam’ rating
  - Repeat..

Graham-Cumming cont.

- End result:
  - Derived a set of key words to turn a spam into ‘ham’.
  - An attack specific to **one** filter config.
- Required 10,000 messages
- Only useful in organizational attack:
  - 10 messages to 1000 people.
- Cost of attack too high for spammers

Challenges

- Filtering becoming an arms race
- A number of issues in defending against attacks:
  - Must test against new variants
  - Different usage scenarios to account for
  - Feedback mechanisms - must be used correctly

Challenges cont.

- Hard to develop (good) attacks:
  - Must keep message intent clear to users, but unclear to filters
  - A black box problem
  - Differing filter configurations
  - Gain v. Effort

Outline

- Introduction
- Background
- Attacking Filters
- *Testing a New Attack*
- Conclusion

Testing a New Attack

- Tested two types of attacks:
  - Dictionary word attack (old)
  - Common word attack (new)
- Both attacks add *n* random words to a base message.
- Tested against two filters:
  - CRM114 - Sparse binary polynomial
  - SpamBayes (SB) - Naïve bayesian

Procedure

- Training data
  - 3000 hams from SpamAssassin corpus
  - 3000 spams from SpamArchive-mod corpus
  - CRM114 trained on errors
  - SB using bulk training

Procedure cont.

- Test data
  - Started with a base ‘picospam’ not in training data:

```
From: Kelsey Stone <bouhooh@entitlement.com>
To: submit@spamarchive.org
Subject: Erase hidden Spies or Trojan Horses from your computer

Erase E-Spyware from your computer

http://boozofoof.spywiper.biz
```

Procedure cont.

- Test data cont.
  - Base picospam is detectable by filters
  - Generated 1000 variations with *n* words added.
    - Words selected with and without replacement
    - n = 10, 25, 50, 100, 200, 300, 400
  - Recorded classifications, effect on score

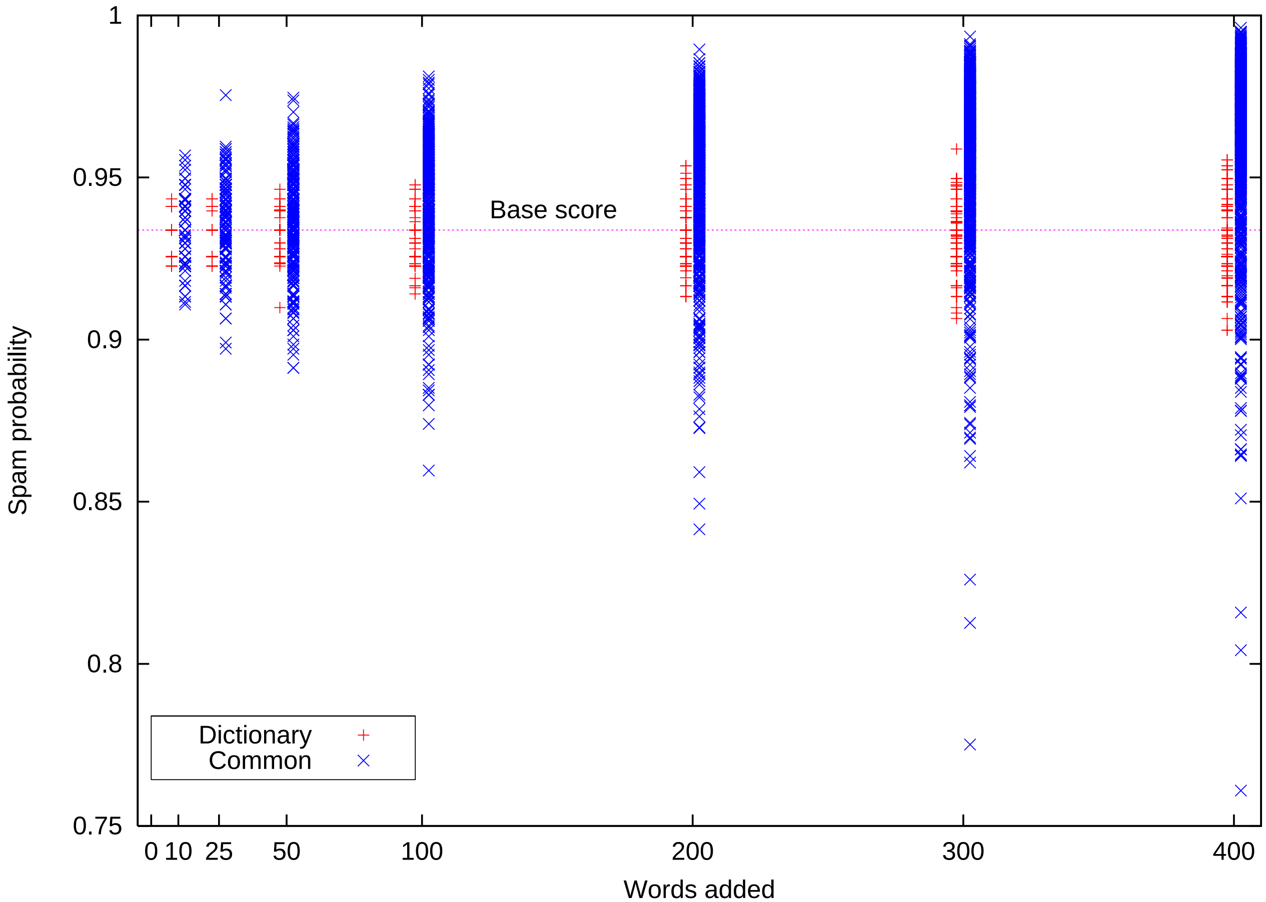
Results

- Using 10,000 variants didn't effect results
- Selection with/without replacement had no effect
- Mixed results

CRM114 Results

- Both attacks failed; 0 false negatives
- Spam score *was* effected...

CRM114 Results cont.



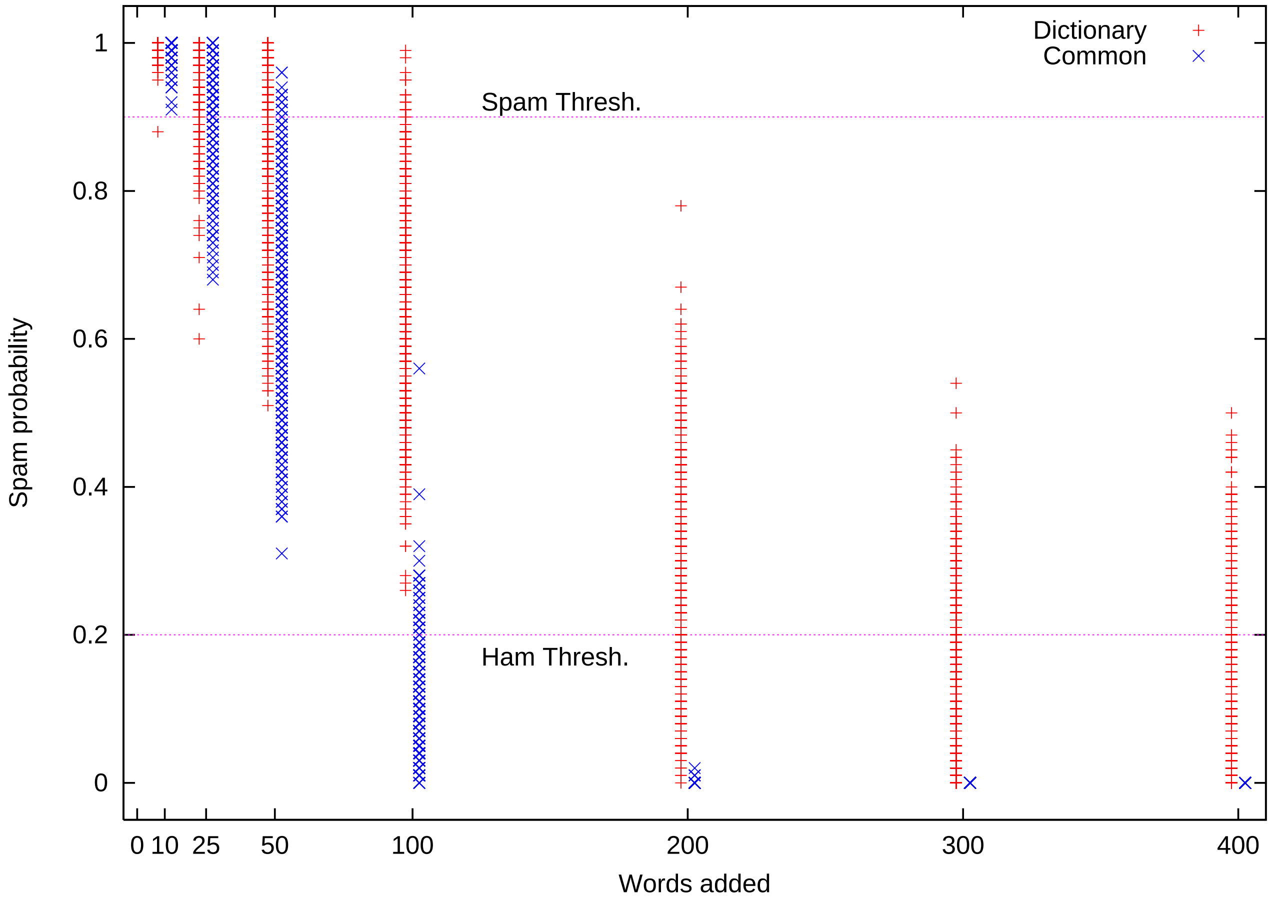
SpamBayes Results

- Baseline Dictionary attack: mild success

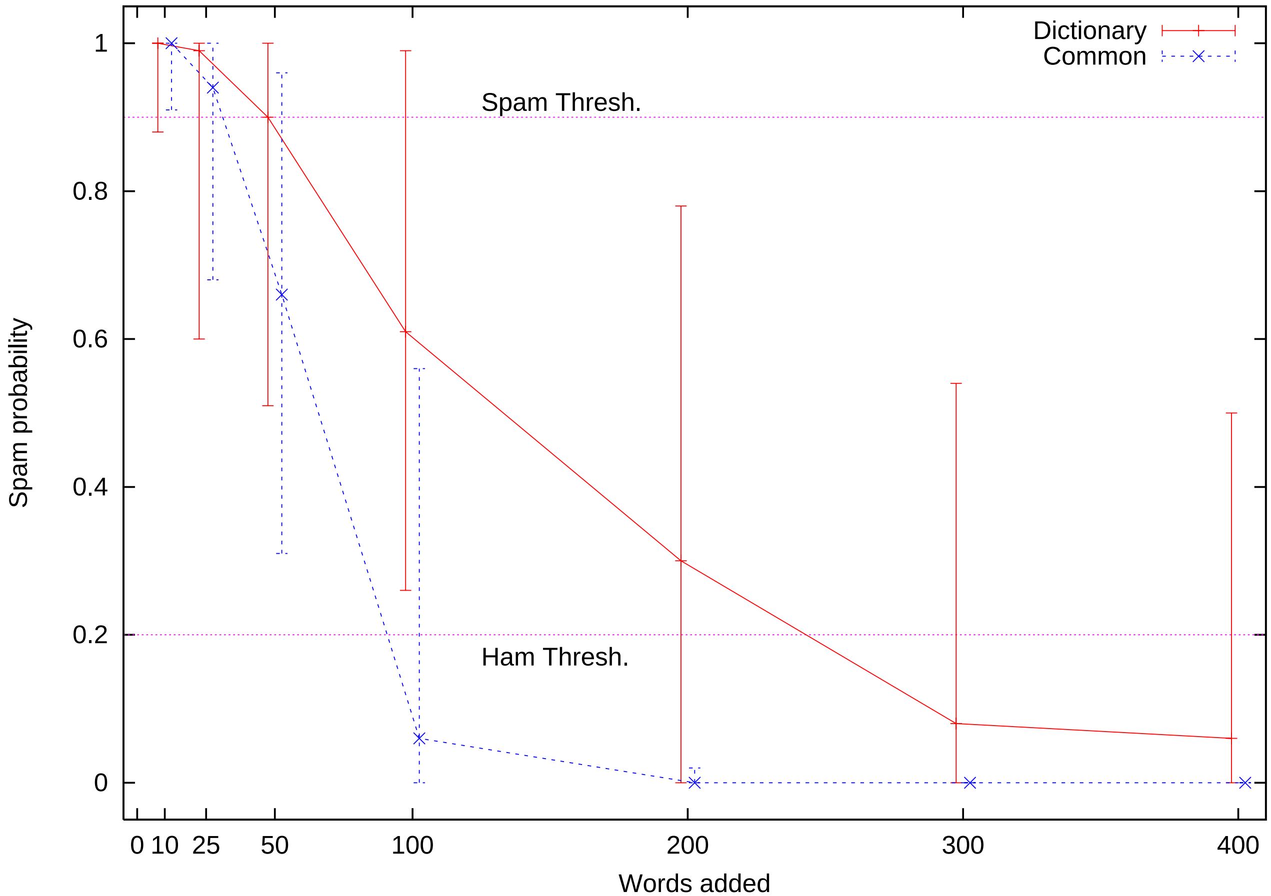
Words	Spam	Ham	Unsure
10	999	0	1
25	937	0	63
50	484	0	516
100	22	0	978
200	0	269	731
300	0	829	171
400	0	858	142

- Common word attack: SB Breaks...

SpamBayes Results cont.



SpamBayes Results cont.



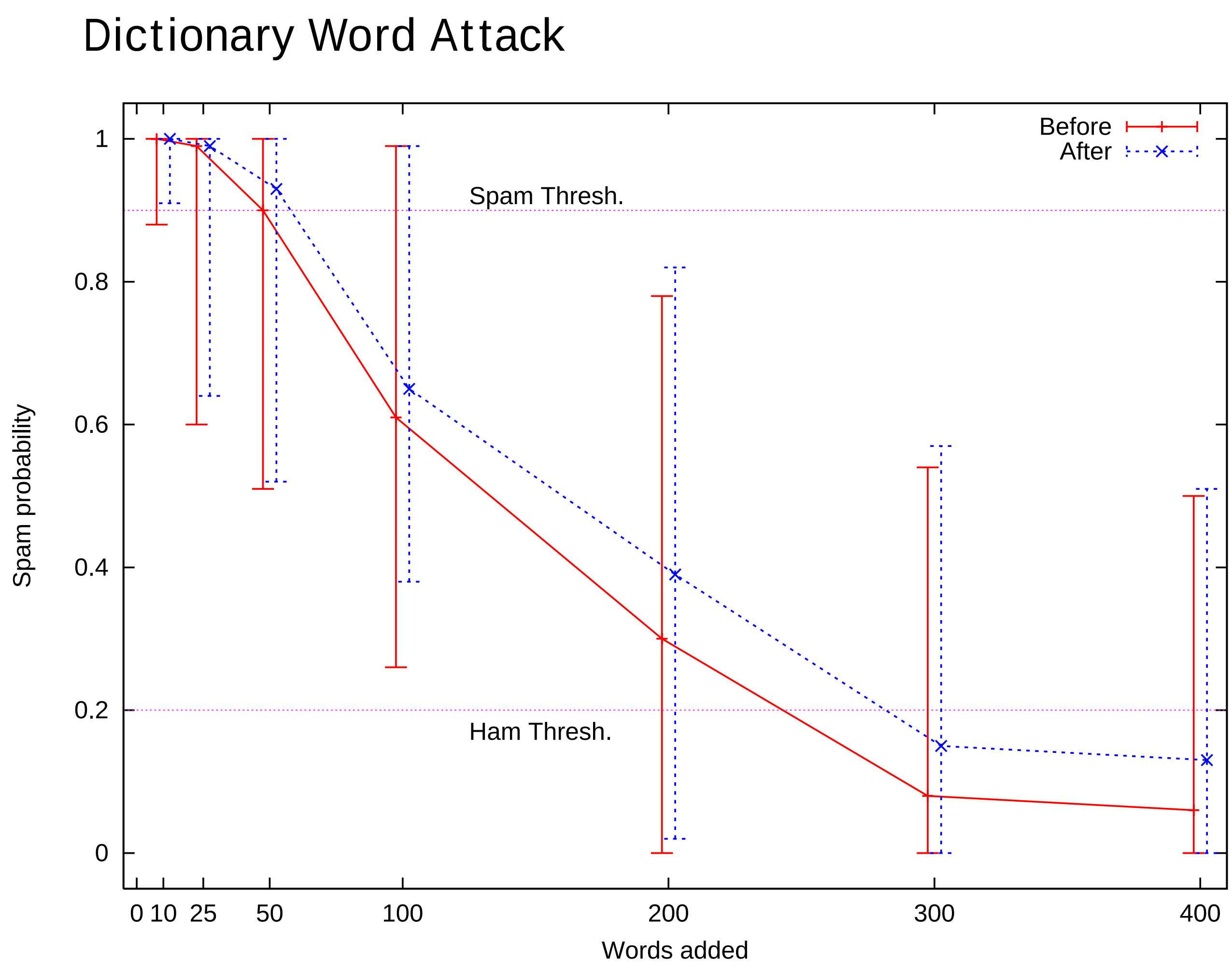
SpamBayes Results cont.

- Common word attack reduces attack size by up to 4x
- Why such poor performance on either attack?
- Hypothesis: Basis picospam was not in training data.
- Added the basis spam to SB’s training data…

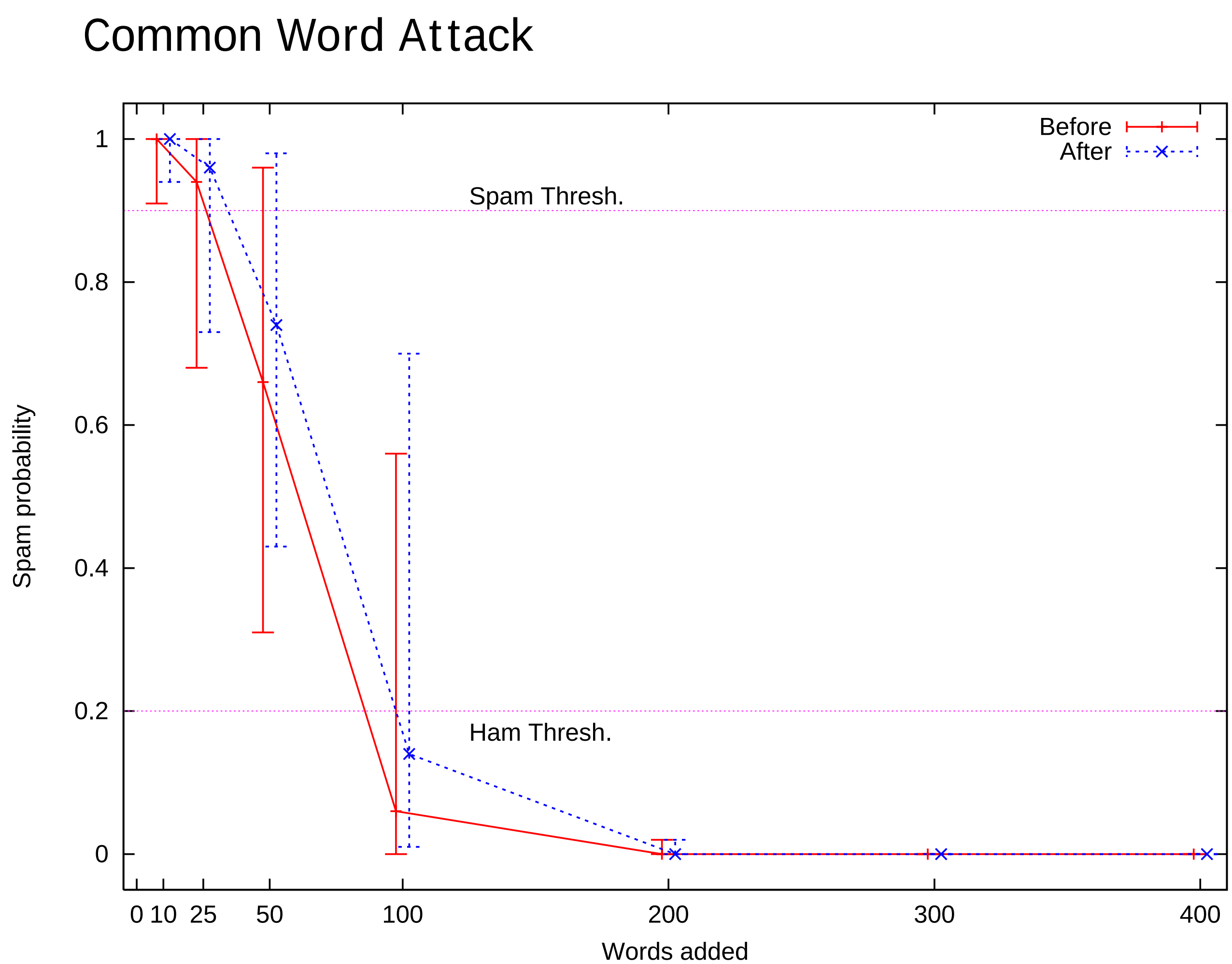
SpamBayes Results Part 2

- Retrained filter offered greater resistance to ‘weak’ dictionary attack.
- Small performance gain against common word attack.

SpamBayes Results Part 2 cont.



SpamBayes Results Part 2 cont.



Outline

- Introduction
- Background
- Attacking Filters
- Testing a New Attack
- *<u>Conclusion</u>*

Conclusion

- Mixed success of common word attack shows need for further study
- Other filters
- Effect of re-training on attack msgs v.
  - False negative, false positive rate
- Testing other basis picospams

Future

- What makes a filter hard to distract?
- More advanced attacks
  – Natural language generation
- Traditional software flaws
  – Exploitable buffer overflows
  – Remote code execution

Further Information

- IRTF Anti-Spam Research Group:
  - http://asrg.sp.am/
- Reading list:
  - http://wwwcsif.cs.ucdavis.edu/~wittel/research/references.html