RescoreDetails320

Rescore Mass-Check Historical Doco for 3.2.0

(see RescoreDetails310 for historical 3.1.0 mass-check documentation, and RescoreMassCheck for the overall rescoring procedure including server-side bits. This is the documentation used for 3.2.0.)

Here's the procedure you'll need to follow, if you wish to submit data for the rescoring run for 3.2.0 using MassCheck:

Clean up the corpus of mail you intend to MassCheck (see CorpusCleaning). The 'checking for false positives and false negatives' stage of corpus cleaning can be done after mass-checks complete, if you like.

Get an rsync account (see RsyncAccounts). If you are submitting nightly mass-check results, the account you use for that will work. Otherwise, getting an account can be done while mass-check is running, since it's not needed until the end.

It's helpful, but not required, to have some or all of the helper applications installed:

- the Mail::SPF::Query module
- the Net::DNS module
- Mail::DKIM module
- Mail::DomainKeys module
- Razor
- Pyzor

If you're running nightly mass-checks, please feel free to disable them when running the rescore mass-check runs.

Note that it's essential that you mass-check *both* ham and spam for this run, as otherwise the Bayes rules will be affected.

What To Run

Then run these commands:

```
wget http://people.apache.org/~jm/mcsnapshot.tgz
tar xvfz mcsnapshot.tgz
cd mcsnapshot
perl Makefile.PL < /dev/null
make

cd masses
mkdir spamassassin
rm -f spamassassin/*
echo "bayes_auto_learn 0" > spamassassin/user_prefs
echo "lock_method flock" >> spamassassin/user_prefs
echo "bayes_store_module Mail::SpamAssassin::BayesStore::SDBM" >> spamassassin/user_prefs
echo "use_auto_whitelist 0" >> spamassassin/user_prefs
echo "whitelist_bounce_relays example.com" >> spamassassin/user_prefs

nohup ./mass-check --progress --bayes --net -j 4 --restart=400 --learn=35 --reuse \
      --after=1072933200 <targets>
```

Explanation

`<targets>` is the list of directories, mboxes, etc., like `spam:dir:~/Mail/spam`. See the comments at the top of "mass-check" for details.

Do not use `--reuse` if you have scanned with SA, but have configured that scanner to run with -L, or you have disabled common network tests or SPF. This is because it relies on the presence of the `X-Spam-Status` line to pick up hits on those rules, and currently cannot detect those conditions.

This takes *ages* to run. `-j 4` controls the number of processes to use; 4 should be OK for a single-processor machine, since most of the time they'll be waiting for network results to arrive. If you have adequate RAM and don't mind the load, you can use `-j 6` or `-j 8`. There's not much benefit in going higher than `-j 8`.

The `--after=1072933200` option tells mass-check to ignore messages older than a certain cut-off point (in this case January 1 2004). This is useful if your corpus has older messages intermingled with your newer messages. This should be old enough to include plenty of good non-spam mail (which doesn't go out of date as quickly as spam) – we will be removing too-old spam in a separate step, anyway, so don't worry about spam.

If you have an unusual network layout, you may need to specify `trusted_networks` and/or `internal_networks` in the `spamassassin/user_prefs` file. But SA should be able to infer it in most cases. A good way to tell is if you see no SPF_PASS results – SPF will not be used if the message passes through one or more trusted relays.

`whitelist_bounce_relays example.com` is an (optional) bit of configuration, which will highlight bounce messages in your corpora. You probably don't want these in your corpora – unless you're certain they're good, non-spam bounces, generated in response to a mail you really did send. Feel free to insert the name(s) of your genuine MSA relays here, if you like, to whitelist the "good" bounces in your corpus.

Once That's Done

Once it finishes, check that the results are sane. See CorpusCleaning to remove any result lines that deal with misclassified or corrupt messages. (This step is very important.)

Then submit your results! You may have to ask for an rsync account at this point, if you haven't already got one set up.

```
  USER="[whatever your username is]"
  RSYNC_PASSWORD="[whatever your password is]"
  export RSYNC_PASSWORD

  rsync -Pcvuzb ham.log $USER@rsync.spamassassin.org::submit/ham-bayes-net-$USER.log
  rsync -Pcvuzb spam.log $USER@rsync.spamassassin.org::submit/spam-bayes-net-$USER.log
```

That's it!

The results for this run will need to be in by Tuesday Feb 6th (tentatively). If you're still running then, submit what you have so far and beg for more time. We may be pushing it out a little further anyway depending on how things go 😉