MassCheck

Using mass-check To Test Rules

"mass-check" is a tool included in the 'masses' directory, which can be found in the SVN repository, to test rules for accuracy and hit-rate. If you're writing custom rules, you really should use this to test them.

First, you need HandClassifiedCorpora. Let's say that's made up of two mbox folders, "/path/to/ham" and "/path/to/spam".

Next, cd into the "masses" directory of the source distribution:

```
cd masses
./mass-check --progress \
          ham:mbox:/path/to/ham \
          spam:mbox:/path/to/spam
```

This will create two files, "ham.log" and "spam.log" containing the hitting rules, read from the rules dir "../rules" as they are applied to that corpus. Each line of the two log files represents details about one email message, and there's a line for every message.

mass-check also takes other options to control whether network tests are run, whether multiple processes are run in parallel, how the output is presented, etc.; read the comments at the top of the file for details. Here's some key bits:

Configuration File

Mass-check reads a "user_prefs" file in "spamassassin/user_prefs". You need to create this yourself, it will not be created for you.

To test your own rules, you'll need to put them in this file, and include a line containing "allow_user_rules 1"

Using network tests

For mass-checks for scoresets 1 or 3, using network tests, you need to provide the `--net` switch. Ensure Net::DNS, Mail::SPF, Mail::DKIM (at least 0.31, preferably 0.36_5 or later), Razor (InstallingRazor), Pyzor (InstallingPyzor) and DCC (InstallingDCC) are installed.

Network tests are slow unless you use the -j switch to allow mass-check to start multiple parallel scanning processes.

Using Bayes

This is controlled using the mass-check configuration file. Do this:

```
cd masses
mkdir spamassassin
rm spamassassin/bayes*
echo "use_bayes 1" >> spamassassin/user_prefs
```

or to turn it off:

```
cd masses
mkdir spamassassin
echo "use_bayes 0" >> spamassassin/user_prefs
```

Once mass-check completes

If you're using mass-check to test your own rules, the next step is to run hit-frequencies: see HitFrequencies for details. Alternatively, if you're submitting data for a new scoreset, see RescoreMassCheck, or NightlyMassCheck for the nightly QA test.

Usage

mass-check [options] target ...

-c=file	set configuration/rules directory
-p=dir	set user-prefs directory
-f=file	read list of targets from <file>
-j=jobs	specify the number of processes to run simultaneously
--net	turn on network checks!
--mid	report Message-ID from each message
--debug	report debugging information
--progress	show progress updates during check
--rewrite=OUT	save rewritten message to OUT (default is /tmp/out)
--showdots	print a dot for each scanned message
--rules=RE	Only test rules matching the given regexp RE
--restart=N	restart all of the children after processing N messages
--deencap=RE	Extract SpamAssassin-encapsulated spam mails only if they were encapsulated by servers matching the regexp RE (default = extract all SpamAssassin-encapsulated mails)

log options

-o	write all logs to stdout
--loghits	log the text hit for patterns (useful for debugging)
--loguris	log the URIs found
--hamlog=log	use <log> as ham log ('ham.log' is default)
--spamlog=log	use <log> as spam log ('spam.log' is default)

message selection options

-n	no date sorting or spam/ham interleaving
--after=N	only test mails received after time_t N (negative values are an offset from current time, e.g. -86400 = last day) or after date as parsed by Time::ParseDate (e.g. '-6 months')
--before=N	same as --after, except received times are before time_t N
--cache	Use cached information about atime (generates files in corpus area)
--all	don't skip big messages
--head=N	only check first N ham and N spam (N messages if -n used)
--tail=N	only check last N ham and N spam (N messages if -n used)

simple target options (implies -o and no ham/spam classification)

--dir	subsequent targets are directories
--file	subsequent targets are files in RFC 822 format
--mbox	subsequent targets are mbox files
--mbx	subsequent targets are mbx files

Just left over functions we should remove at some point:

--bayes	report score from Bayesian classifier

Usage: Targets

non-option arguments are used as target names (mail files and folders), the target format is: <class>:<format>:<location>

class	is "spam" or "ham"
format	is "detect", "dir", "file", "mbx", or "mbox"
location	is a file or directory name. Globbing of ~ and * is supported.

"detect" is the easiest format to use. This assumes "mbox" for any file whose path contains the pattern "/\.mbox/i", "directory" for anything that is a directory, or "file" otherwise.