Appendix D: Data Collection Scripts

The following scripts are used to collect twitter messages and store them in a format accessible to both python and unicage. While a databasing system has its obvious advantages, this methodology is the paragon of simplicity.

Curl URL Builder

This script generates a properly signed URL for opening a twitter stream via curl

```
""" twitter_curl_url_builder.py """

import oauth2 as oauth
import time

# Set the API endpoint
url = 'https://stream.twitter.com/1.1/statuses/sample.json'

# Set the base oauth_* parameters along with any other parameters required
# for the API call.
params = {
    'oauth_version': "1.0",
    'oauth_nonce': oauth.generate_nonce(),
    'oauth_timestamp': int(time.time())
}

# Set up instances of our Token and Consumer.
token = oauth.Token(key='**************************',
                    secret='*************************')
consumer = oauth.Consumer(key='**********************',
                          secret='********************')

# Set our token/key parameters
params['oauth_token'] = token.key
params['oauth_consumer_key'] = consumer.key

# Create our request. Change method, etc. accordingly.
req = oauth.Request(method="GET", url=url, parameters=params)

# Sign the request.
signature_method = oauth.SignatureMethod_HMAC_SHA1()
req.sign_request(signature_method, consumer, token)

print req.to_url()
```

To use, at the command prompt:

```
shell
$ URL=$(python twitter_curl_kickstarter.py)
$ curl -get "$URL"
```

Twitter Stream Opener

This script starts a curl process to get posts from twitter and saves them to 100000 post long files. We use the

python script `twitter_curl_url_builder.py` to handle the oauth bits, as they can be complicated in bash.

```
#!/bin/bash

URL=$(python twitter_curl_url_builder.py)

curl --get "$URL" | split -l 100000 - ../data/posts_sample_`date "+%Y%m%d_%H%M%S"`_

echo "`date` Twitter stream broken with error: ${PIPESTATUS[0]}" >> tw_collect_log.txt

# the curl should go on indefinitely, so if we get to this point, an error has occurred, raise
exit 1
```

Twitter Stream Monitor

This script starts the stream, and watches to see if it fails. If so, it restarts the process after some amount of time.

For a great description of the watchdog loop, see:
http://stackoverflow.com/questions/696839/how-do-i-write-a-bash-script-to-restart-a-process-if-it-dies

```
#!/bin/bash
reconnect_delay=600

until ./twitter_stream_opener.sh; do
    echo "`date` Twitter curl process interrupted. Attempting reconnect after $reconnect_delay

    echo "`date` Twitter curl process interrupted. Attempting reconnect after $reconnect_delay

    sleep "$reconnect_delay"

done
```