Introduction to Digital Libraries Assignment #1

James Tate II

February 11, 2015

1 Introduction

This assignment required downloading *tweets* from *Twitter* using its API and processing the HTTP URIs those tweets contained. The URIs were dereferenced from their original http://t.co/... form to their final URI after following all redirects. The representation obtained by dereferencing the final URI was recorded for later use. The URIs were also given an age by the CarbonDate utility. This report includes some statistical data gathered along the way.

2 Methodology

The data for this assignment was obtained and processed in several stages rather than all at once. This enabled agile development, and also did not require much effort be put into usability or maintainability of the code used at each stage. Six significant Python scripts were developed, along with a library of miscellaneous functions not worth mentioning. Three Bash scripts were also developed. Third-party resources included CarbonDate, casperjs, phantomjs and CherryPy, along with the Python standard library, wget, curl and numerous other GNU utilities.

2.1 Downloading Tweets

The first was, of course, to download enough tweets that 10000 URIs were able to be analysed after culling the unusable tweets. Tweets were downloaded using Twitter's streaming API. A list of 23 keywords were used in the `filter` endpoint on the streaming API, as special permission is required to use the firehose endpoint. Because the streaming API was used, retireved tweets were newly published tweets at the time of retrieval. The JSON of each tweet was saved to an output file for later processing. The command to download tweets is shown in listing 2-1. The output file is always output.log.

Listing 2-1: Downloading Tweets

```
./download_tweets.py
```

2.2 Culling Tweets

The first step in culling tweets was a naïve attempt to omit tweets with adult content. grep was used ias shown in listing 2-2 to remove tweets containing the string porn in any case, regardless of position in a word.

Listing 2-2: Removing Naughty Tweets

```
grep -vi "porn" output.log > output.log.2
```

Appendices

A Streaming API Filter Keywords

These keywords were selected arbitrarily. Keywords were added to the list until the streaming API seemed to pull tweets at a strong, consistent rate.