Introduction to Digital Libraries Assignment #4

James Tate II

April 23, 2015

1 Introduction

Assignment #4 first required analyzing the changes (if any) in the representations retrieved in assignment #3. The same URIs were dereferenced again and compared to the previous representations. The next part of this assignment involved retrieving TimeMaps of the analyzed URIs, and analyzing the number of mementos each URI had. The last part of the assignment required generating graphs of Jaccard Distance over time of 20 URIs.

2 Methodology

I wrote several scripts for this assignment and I also modified some previously-written scripts. They are described in subsections for the different questions in this assginment.

2.1 Question #1

The first part of Question #1 was to again dereference the URIs that were successfully processed by boilerpipe in Assignment #3[2]. I used the command in Listing 2-1 to get a list of the dereferenced URIs that were successfully processed in Assignment #3.

Listing 2.1: Getting URIs that Succeded Boilerpipe Processing

```
grep -v " 0 " wc_boilerpipe | awk4 \
  | grep -o "\./tweets/[0-9]\+/" \
  > boilerpipe_success_dirs
```

Next, I created a new directory to store the second representations of the URIs. I populated the directory will the URI files I would need to dereference the URIs. These steps are in Listing 2.2.

Listing 2.2: Creating Second Tweets Directory

```
mkdir tweets2
for i in $(cat boilerpipe_success_dirs);
  do id=$(echo $i | grep -o "[0-9]\+");
  mkdir tweets2/$id;
  cp "${i}url.0" tweets2/$id/url.0;
done
```

I then dereferenced the URIs using the same script from Assignment #1 which I modified to used the new directory[1]. This step is shown in Listing 2.3. It took 136 seconds to dereference 3035 URIs using 128 parallel processes. 146 did not dereference successfully.

Listing 2.3: Dereferencing URIs

```
./dereference_URIs.py
```

After obtaining new representations, I created a list of downloaded represenations and processed the files with boilerpipe, as shown in Listing 2.4.

Listing 2.4: Extracting Text with Boilerpipe

```
find ./tweets2/ > tweets2_file_list
./run_boilerpipe.py
```

Once I had the textual output from boilerpipe I was ready to start the processing with my *jaccard.py* script. This script removes most punctuation from the text, generates sets of unigrams, bigrams and trigrams then calculates the Jaccard Distance between the two representations of each resource[3]. These commands to setup input lists for my script and to run in are in Listing 2.5. The first four commands removed URIs with a second representation that was not successfuly processed by boilerpipe.

Listing 2.5: Calculating Jaccard Distance

```
wc tweets2/*/*/boilerpipe.output | tee wc_boilerpipe2
grep -v "^ \+0 " wc_boilerpipe | grep -v " total$" \
  | awk4 | grep -o "56[0-9]\+" > boilerpipe1_ids
grep -v "^ \+0 " wc_boilerpipe2 | grep -v " total$" \
  | awk4 | grep -o "56[0-9]\+" > boilerpipe2_ids
comm -12 <(sort boilerpipe1_ids) <(sort boilerpipe2_ids) \
  > boilerpipe_common_ids
./jaccard.py | tee hw4_report/stats/q1_distances.stats
```

The last step for this question was to generate three files for consumption by R to make the graphs in the Results section. These files were generated by the commands in Listing 2.6.

Listing 2.6: Generating R Input Files

```
awk2 q1_distances.stats | tail -n +2 \
  | sort -n > q1_unigrams.stats
awk3 q1_distances.stats | tail -n +2 \
  | sort -n > q1_bigrams.stats
awk4 q1_distances.stats | tail -n +2 \
  | sort -n > q1_trigrams.stats
```

2.2 Question #2

2.3 Question #3

3 Results

3.1 Question #1

3.2 Question #2

3.3 Question #3

Table 3.1: Word Count Data

Data Point	**Original**	**After jusText**
Total bytes	646,619,835	12,133,748
Total words	33,594,568	2,035,935
Unique words	9,135,191	121,422
Total letter words	81,318,873	2,030,636
Unique letter words	1,271,593	57,434

References

[1] James Tate II. CS 751 Assignment #1. 2015.

[2] James Tate II. CS 751 Assignment #3. 2015.

[3] Wikipedia. Jaccard index — wikipedia, the free encyclopedia, 2015. [Online; accessed 24-April-2015].