Assignment 4 : CS 751

HARISH

May 1, 2015

0.1 Question 1

Using the pages from A3 that boilerpipe successfully processed, download those representations again and reprocess them with boilerpipe.
Document the time difference
Compute the Jaccard Distance x for each Pair of Pages for :Unique terms ,Bigrams,TriGrams
For each of 3 cases build a Cumulative Distribution Function that shows the percentage change on the x-axis and percentage of the population on y-axis
Give 3-4 examples illustrating the range of change that you have measured

Solution to Question1:

I have used jusText library to remove HTML templates in Assignment 3 and 4. It removes headers, footers and navigation links from the page. jusText was successful for pages that followed HTML standards. Pages which are properly organized .

I have written a code in python named "htmlnonhtml.py" which is located in "q1-A4" folder. This code removes boilerplate content for the links present in "responseAfterCurlCalls" located in "q1-A4" folder.

Only 6100 URIs were unique , out of this only for 1496 URIs boilerpipe sucessfully processed . Most of them are non-HTML skipped because of non content , Images on links , 404, 302 errors etc.

I have used the same pages from A3 where boilerpipe was successfully processed. I have reprocessed those pages with boilerpipe and extracted the text files again for A4

The time difference between A4 and A3 is

Time(A4)-Time(A3) : 28 days

I have created two folders "A3-afterboilerplateTextFiles", "A4-afterboilerplatetextfiles" which is located inside "q1-A4" folder. These folders contains text files that were extracted after processing boilerpipe successfully for A3 and A4.

I have calculated Unique terms ,Bigrams , Trigrams for the text files located in the following folders :

1)"A3-afterboilerplateTextFiles" located inside "q1-A4" folder
2)"A4-afterboilerplatetextfiles" located inside "q1-A4" folder

I have written a java code named "UniGram.java","BiGram.java", "TriGram.java" located in "q1-A4" folder to extract unique term , bigrams and trigrams respectively for the text files .

I wrote a "jacarrdDistanceCalculator.py" code located inside "q1-A4" folder . This code reads the textual data generated for each pair of pages from A3 and A4 and calculates the jaccard distance for unique terms , bigrams and trigrams and saves them to an output file named "unijackard-sorted.txt" , "bijacardsorted", "trijacardsorted" respectively. These files are located inside the folder "q1-A4"

For each of the 3 cases (i.e., 1-,2-,3-grams) build a cumulative distribution function that shows percentage change on the x-axis and the percentage of population on the y-axis.

Figure 1 ,Figure 2 , Figure 3 represents the Cumulative Distribution Function for Unigram , BiGram , TriGram respectively . I have plotted the graph using R script . BiJacard.R , UniJacard.R , TriJacard.R are the scripts for generating Cumulative Distribution Function graphs for Bigrams , UniGrams, Trigrams respectively . These are located inside "q1-A4" folder.

Following are the three examples illustrating the range of change measured

1) For this link "http://pinterest.com/pin/300896818829344973/" after successfully processing boilerpipe , there is a lot of change in textual content . Interesting part here is that , the jaccard distance for unigram for this link is "0.958333333333" where as for Bigram and TriGram it is "1"

2) For this link ”http://suba.me” after successfully processing boilerpipe , there is a little change in textual content . But, the jaccard distance for unigram , bigram and trigram are 0.0173611111111 ,0.0138248847926 ,0.0134831460674 respectively . Jaccard Distances are not same for unigram , bigram and trigram

3) For this link ”http://twi.xsrv.jp/eot4j” after successfully processing boilerpipe, there is no change in the textual content. The Jaccard distances(1-gram, 2-gram, 3-gram) for these pair of pages from A3 and A4 is 0.

0.2 Question 2

from Q1 (A4), download all TimeMaps(including TimeMaps with 404 responses, i.e. empty or null TimeMaps)
Upload all TimeMaps to github
Build a CDF for number of mementos for each original URI i.e., x-axis = number of mementos , y-axis = number of links

Solution to Question2:
I have written a code ”getURLData.py” which is located inside the ”q2-A4” folder. This code reads ”urlStatusAfterCurlCalls” file which contains the URIs. This is located in ”q2-A4” .
I have used ”http://labs.mementoweb.org/timemap/link/” to get the archived data using wget . This codes saves all the TimeMaps for all the 10000 links. These are saved inside ”q2-A4/timemaps/” folder .This list includes the null time maps as well.

After collecting the mementos for each URI , I have written a python script named ”mementocount.py” which extracts the count of mementos per each URI and saves it to ”timemapsCount.txt” file which is located inside ”q2-A4” folder.

Figure 4 shows the cumulative distribution function for all the count of mementos . I have written a R script named "cdfformemntos.R" located inside "q2-A4". Most of the URIs have zero mementos.

0.3 Question 3

Using 20 links that have TimeMaps -
-With¿=20 mementos
-Have existed greater than equal to 2 years i.e., Memento-Datetime of "first memento" is April XX, 2013 or older
-Note : select from Q1/Q2 links , else choose them by hand
For each link , create a graph that shows Jaccard Distance, relative to the first memnto, through time
-x-axis : continuous time ,y-axis: Jaccard Distance relative to the first memento

Solution to Question3:
I have chosen 20 links by hand that satisfies the question criteria. I have saved these files into "final20links.txt" which is located inside "q3-A4" folder. I have written a code "getURLData.py" located inside "q3-A4" folder . This code reads "final20links.txt" and extracts the mementos links for each URI in the list in json format with the help of "http://labs.mementoweb.org/timemap/json/" .

I have saved all of these files with '.json' extension. I have written a python script with name "html-non-html.py" which reads each json file in the directory ("q3-A4" folder) and iterates through each memento present in the list (json Array) for each URI and applies boilerpipe to each link. Most of the links were not successfully processed after applying boilerpipe. For the URIs where boilerpipe successfully got processed , I have calculated the Jaccard distance relative to first memento.

I have written a Rscript named ”JDRelativetofirstMemento.R” which is located in ”q3-A4” folder

so and so figures represent nvksdvn

0.4 Question 4

Choose a news-related event
UseTwarc.py to collect 1000 tweets , every day for 5 different days
For each day:
- Create a wall
- Build a tag/word cloud for each day
- Create a map using GeoJSON and Github
Discuss in detail lessons learned , experiences , etc

Solution to Question4:

First ,I have generated a consumer key , consumer secret key , access token and access-secret token.
I have collected 1000 tweets , every day for 5 different days.
I have written a script(using above mentioned keys) named ”twarc1.py” located inside folder ”q4-A4”. I have extracted the tweets on ”AppleWatch” tag . I have saved these 1000 tweets daily with ”.json” extension .
The files are located inside the folder ”q4-A4”.Following are the files I have extracted on a daily basis for 5 different days

tweets25th.json
tweets26th.json
tweets27th.json
tweets28th.json
tweets30th.json

I have copied all of these inside ”twarc-master/utils/” folder which is located in ”q4-A4” folder .

I have given above json files as an input to wall.py to create Wall on daily basis. It gave me output wall with ’.html’ extension. All of these output files are stored inside ”q4-A4/wall/” folder.

I have given above files as an input to word.py to create word cloud on daily basis. It gave me output word cloud files with ’.html’ extension. The files are located inside the folder ”q4-A4/wordcloud/”.

I have also given above json files as an input to tweets.geojson to get geo coordinates from the tweets.