Data Management UNIX Coursework

Brody Walters, (bw2u18), Student ID: 30422817

March 2019

1 Scripts

1.1 Countreviews.sh

```
#!/bin/sh

for file in $1/∗;  do
name=$(echo $file | awk −F”/” ’{print $NF}’ | sed ”s/.dat//g”)
reviews=$(awk ’/<Author>/{sum += 1} END{print sum}’ $file)
echo $name $reviews
done | sort −k2 −n −−reverse
#EOF
```

This script is used to count the number of reviews a particular hotel has received.

1.) for file in $1/*; do

A for loop that executes for each file in a given directory.

”$1” is used to represent the first positional argument. Name is used on the next line as variable which outputs the first part of the script. This is done by separating the file path ’/’ and uses the last column (hotelname.dat extension), this is pipe-lined and sed is used to replace ”.dat” with a null string. This leads the string to be to hotel ”hotel number”, where each hotel has an individual hotel number.

2.) reviews=$(awk ’/¡Author¿/sum += 1 ENDprint sum’ $file)

Awk is used here to scan a particular file from the for loop and search for the string ”Author”, with 1 being added to the count every time it is found. Once the file has been scanned, the END command is used to print the sum. The output is then allocated to the variable ”reviews” which is the number of reviews a particular hotel has. The echo command is then used to print the ”name” variable followed by the ”reviews” variable. The output of this is pipe-lined to the sort command which sorts by the size of the ”reviews” number and displays the result in descending order.

2 Scripts

2.1 Average reviews script

```
#!/bin/sh

cd ”$1” || exit 1

for file in ∗.dat; do
  (printf ”%s ” ”$file” | sed ”s/.dat//g”)
  awk ’/<Overall>/ {sub(/<Overall>/, ””); count += 1; sum += $0}
 END {printf ”%.2f\n” ,sum/count}’ ”$file”
done | sort −k2 −n −−reverse
```

This script is used to rank hotels by their average review rating and display them in descending order.

1.) for file in *.dat

A for loop to loop over each file individually, this is done by using the * wildcard.

2.) (printf ”%s ” ”$file” — sed ”s/.dat//g”)
This prints the name of the hotel with the output pipe-lined to sed which removes the ”.dat” extension on hotel files

3.) awk ’/¡Overall¿/ sub(/¡Overall¿/, ””); count += 1; sum += $0
Awk is used here to scan for the key word ”Overall” as this precedes the review rating of the hotel. Sub is used to replace ”¡Overall¿” with a blank space so that only the number of the rank is fetched. Count is used to represent the number of reviews a hotel as been given and sum is the total of all the review ranks added. ”$0” is used for sum as the review rank is the first positional parameter following ”¡Overall¿”.

4.) END printf ”%.2f’ ,sum/count’ ”$file”
This prints ”$file” which is the name of the hotel , followed by the result of sum/count which is the average rating of review of the hotel. ”%.2fis used to print the average rating of the hotel to 2 decimal places.

3 Discussion

3.1 Structured database vs. Unstructured mark up

Our task essentially was to extract data from various unstructured mark up documents. The most efficient way to extract this data was to use awk to serach for key words in the data to find what we were looking for. The files were arranged in the same ways which made it easier to scan for key words through files. If the hotel reviews where to be files from 2 different companies, the script would likely not work as the companies would have differing methods of displaying the data. Relational databases would have been easier to extract data from for this coursework as you would only need to use keys to search for data.

3.2 Authenticating reviews

Here are some methods which could be used to ensure that reviews are authentic, rather than fake reviews that give a false representation of a particular hotel.

1.) Access to reviewers' data

The account of the author that post reviews could be accessed to determine the likelihood that the review is authentic. Their review history could be examined to see if the reviewer uses similar language or shows any extreme biases to certain companies.

A superior method to ensure reviews are authentic would require access to private data about the reviewer. It could be examined whether reviews are being posted by different users from the same IP or MAC address. This would be very suspicious and is likely to indicate the reviews are not genuine.

2.) Review stats

Furthermore, how long the user spent writing the review could be an indicator whether the review is genuine. For example, if the review is a few paragraphs long, but was written in 10seconds, it is very likely the review is not authentic.

3.3 Improving review ranking system

There are ways the review ranking system could be improved.

1.) Details about reviewer

User information such as age and gender could be extremely helpful. If they were displayed for users, it could allow us to enhance our scripts to determine what hotels may be best for certain age groups. This would increase the chances that customers interested in booking a hotel choose the best option for them.

2.) Ease of leaving a review
Leaving a review on many websites can be a boring process, where you may be forced to write until a certain word limit is exceeded. It would be much more useful if users could simply leave a number ranking. This would increase the number of reviews being left, which would give a more accurate depiction of the hotel experience.

3.4 Data storage issues with flat-file structure

A flat file database is a database that stores data in a plain text file, they are inferior to relation databases for a few reasons.

Relational databases are much faster and easier at accessing data compared to the flat-file format. As the flat-file format is a text file, the structure of the file has to be considered when approaching data analysis. This is much less efficient than simply using keys to look up particular data that is possible with relational databases. An advantage of relational databases is that data can be updated very simply, this will become apparent as the number of reviews left for hotels increase. To analyse new reviews which are stored as flat files, a download would be required. Lastly, a disadvantage of using a flat-file format for hotel reviews is that it can suffer from data redundancy and some data that is stored can be irrelevant.