Tweet Popularity Predictor

Administrative:

Team Name: Tweet Popularity Predictor
Team Members: Kian Bagherlee, Isabel Mitre, Sebastian Stockbridge
Github URL: https://github.com/isamitre/tweet-popularity-predictor
Link to the Video: https://www.youtube.com/watch?v=TFpVD2ZQBT8

Extended and Refined Proposal

Problem: What problem are we trying to solve?

- The problem we are trying to solve is how well received a tweet, or an ad created by advertisers, would be to the general public.

Motivation: Why is this a problem?

- Online marketing has now become absolutely essential for a company's growth. On top of this, because of how much content is posted everyday, the advertisement must not only be direct, but it must be well received to go "viral."
- Instead of wasting resources to determine whether an advertisement tweet would be well liked or not, our program would simply be able to give companies a metric on how good, or bad, their idea is.

Features Implemented

- You can enter any tweet you want, so there is no limit to what you want to test as popular or not.
- You can choose how you want your information processed: with a Hash Table or Red Black Tree. After pressing the button of your choice, you then have to wait until a new screen pops up. This new screen will first offer you how long it took for your answer to be found in seconds, allowing you to compare whether a Hash Table or Red Black tree was faster.
- You can see your tweet's popularity score. A 4 is the best score, a 2 is a neutral score, and a 0 is the worst score.
- You can see which words in your tweet were the most influential to the tweet's overall popularity score. If you have over a 2, then it will show you the most positively influential words. If you have under a 2, then it will show you the most negatively influential words.
- The program offers a visual to understand the score, and allows you a "Try Again" function which will display the starting window again and allow the user to enter another tweet and perform their desired calculations.

Description of Data

- link to public data set
- there are six columns:

    - an int polarity value with a negative value of 0 and a positive value of 4
    - an int ID of the tweet
    - a string date in this format: Sat May 16 23:58:44 UTC 2009
    - The query (*lyx*). If there is no query, then this value is NO_QUERY
    - a string username of the user who tweeted the tweet
    - a string of the actual tweet text

Tools/Languages/APIs/Libraries used

- Java
- Java Swing for the GUI

Data Structures/Algorithms implemented

- Pair class of two generic types (K key and V value),
    - complete with a constructor, two accessor methods, and a toString method
    - we used it specifically to store String and Floats

Data Structures/Algorithms used

- PriorityQueue (min/max heap) and Find Kth Largest Element Algorithm
    - we used these data structures along with the kth largest elements algorithm from the Module 5 slides to find the words that most influence the inputted tweet positively and negatively
- HashMap<String, int[]> and TreeMap<string, int[2]>
    - we used a map data structure, to represent a Hash Table, with a word as the key and an array/vector of 2 elements:
        - the first is the number of tweets that the word has appeared in
        - the second will be the total sum of the popularity predictor number associated with each tweet that the word appears in
- ArrayList<Pair<String, Float>>
    - we used an ArrayList of Pairs to store the most influential words

Distribution of Responsibility and Roles: Who did what?

- Kian: Parse through the database to create a hashmap with each distinct word in any of the tweets in the database, along with an array. This array will have the total polarity sum, and an entry of how many unique tweets contained the word.
- Isabel: take in user data (string that is a tweet), parse it (remove filler words, break up into individual words), calculate popularity score, get the k words that influenced the popularity predictor score the most (either positively for a good score or negatively for a bad score)
- Sebastian: Plan and build the GUI and overall user experience

Analysis

Any changes the group made after the proposal? The rationale behind the changes.

Changes the group made after the proposal was related to the GUI and how to do some of the tasks. We were confused with the initial instructions of the project, and instead of having 2 data structures/algorithms doing the same task and then comparing the two, we had 2 data structures/algorithms where each did a separate task. After it was made clear to us, we decided to use a Hash Table and a Red Black tree as the two data structures to hold the data from the 1.6 million tweets, as compared to just a Hash Table. This also affected what the GUI would look like as well, and the changes were made as well. Another change was using Java Swing rather than JavaFX for the GUI. The rationale was that Java Swing was much easier to work with and learn than JavaFX.

Complexity analysis of the major functions/features you implemented in terms of Big O for the worst case

- Let string operations be of complexity $O(1)$
- Let n be the number of words in the maps
- Let w be the number of relevant (non-stop) words in the user tweet (total characters less than 280)
- Let a be the max number of words with the same average popularity score
- Let m be the number of words in the user inputted tweet
- Let f be the number of entries in the CSV document

getHashScore and getTreeScore: $O(wn)$

getPowerWordsHashList and getPowerWordsTreeList: $O(wn + w^2)$

parseTweet: $O(m)$

removeStopWords: $O(m)$

Setting up TreeMap: $O(fm * log(w))$

Setting up HashMap: $O(fm)$

Reflection

As a group, how was the overall experience for the project?

The overall experience was good: we all learned a lot and worked together well. We all felt that the completed project looks very good, much better than what was originally thought of. We also felt that individually, we all did our parts very well. We had an extremely smooth process from the start to the end, and what resulted is a project we are proud of.

Did you have any challenges? If so, describe.

One of the biggest challenges faced was during the last stage. While putting together all of the code, the GUI was not showing any output, even though all of the output was calculated. Debugging this error required all members of the group collaborating, which resulted in us fixing the error and finishing the project.

If you were to start once again as a group, any changes you would make to the project and/or workflow?

Admittedly, there are no changes we would make. We did a very good job of dividing the labor so everyone has an equal part to work on, while also communicating with each other on our requests. If there was a change one person needed, or requested, the other would respond quickly and prioritize the change. Any questions were quickly answered, and one of the biggest aspects was how we were able to write our code in a way that made all aspects we worked on extremely fast and simple. The combining of all 3 of our parts went unbelievably smooth. Deadlines were also hit on time, so overall there would really be nothing to change. A very positive experience all together.

Comment on what each of the members learned through this process.

Kian:

- I learned how to properly read through a CSV document in java, as well as how to properly parse out all the information, or any information I personally needed.
- I learned how to use TreeMaps and HashMaps in Java, as well as learned a little about JPanels
- I learned about Regular Expressions, and I also learned how to make regex expressions for different scenarios I wanted.

Isabel:

- I learned how to convert knowledge from one programming language to another. We built our project in Java, as that is what we are all generally more comfortable with. However, throughout this course I learned how to use maps/unordered_maps in C++. It took a bit of adjusting to get used to seeing TreeMaps and HashMaps instead.

Sebastian:

- I learned how to build user interfaces in java, which was something that I have been wanting to learn for a while to use in the future for personal projects or other endeavors.
- More generally, I learned how to teach myself new programming tools because building a GUI in java was not covered in any of my classes.

References

1) “Java GUI (NEW)” - Video about how to build a GUI in Java using Java Swing https://www.youtube.com/watch?v=Kmgo00avvEw&t=7067s
2) Professor Kapoor’s Module 5 Slides on Canvas