CS231 Project 6: Word Frequency Analysis

October 23rd, 2023

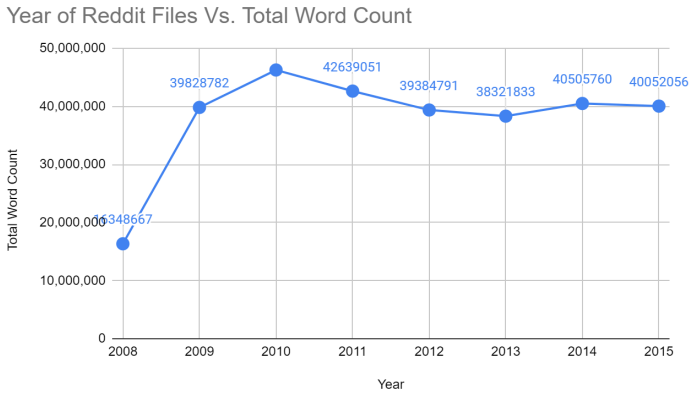
Abstract

This project uses the data structure, Binary Search Tree, to map a set of keys to specific values. The Tree was created from the BSTMap class that contains all the operation methods of a Binary Tree. From this implementation, the word frequencies in Reddit comments over a period of eight years can be analyzed. To analyze the Reddit comments, a WordCounter class was created that manages and initializes the BSTMap of word-count pairs. The WordCounter class was designed to build a BSTMap from a text document, write out the file containing the total word count along with each word and its corresponding count, and finally, read a wordcount file. From the creation of BSTMap and WordCounter classes, a time analysis of computing word frequency could be demonstrated.

Results

For the sake of analysis, all words will be considered

Include a discussion as to whether the results make sense and what they mean.



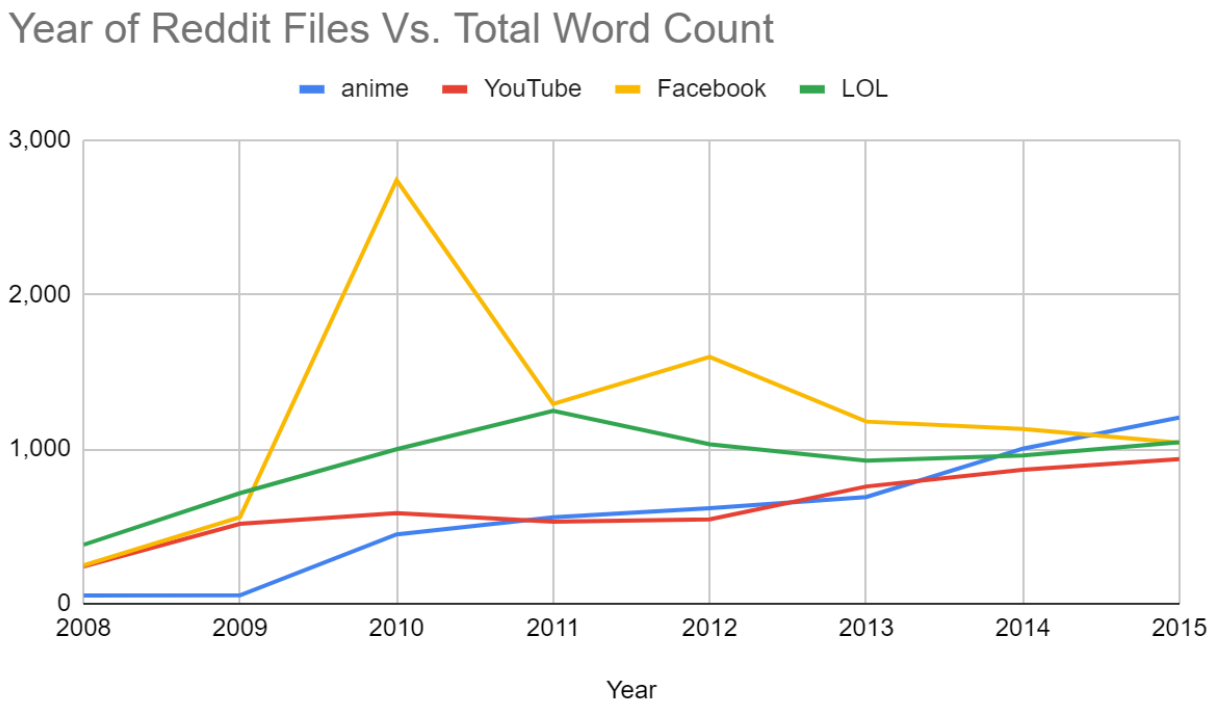
It is expected that the total word count in 2008 would be the lowest, as Reddit slowly became a popular platform that many users would use, thus increasing the word count as the year progresses, when more users log on and post more things.

As for the word counters themselves, it makes sense that common words (such as articles, prepositions, and conjunctions) have the highest usage count. In many contexts, these words are essential for constructing grammatically correct sentences and understanding communication patterns. In the end, reddit comments are just online constructed sentences that still follow the same methodology of making regular sentences.

Exploration

Week 1 results

Pick a few words that matter to you. How do they change year by year? Does it match your expectations?



Generally, as said earlier there should be an expected rise in all words as Reddit slowly becomes more easily accessible to more people. Also one thing to note is that the search algorithm was somewhat primitive as it considered case sensitive words and other special cases such as special characters being in front of the target word being their own thing. For example the words *YouTube, youtube, "YouTube",* and *:YouTube* were all considered their own entries. Because there were too many to locate and add up manually, I considered only what I considered to be the most popular usage and pronunciation of the word, as indicated in the legend.

Anime:

- **Trend**: The interest in anime consistently increased from 2008 to 2015.
- **Significance**: The growing popularity of anime, potentially influenced by international releases and online streaming platforms, is evident.
- **Expectations**: I honestly did not expect anime to have the highest word count by 2015, as Reddit is a platform mostly used by Americans and Eastern Culture of which anime thrives

from is fairly foreign, whereas other more mainstream and well known social media such as YouTube can be enjoyed by many people. However, this progression may not be as significant for reasons highlighted early.

YouTube:

- **Trend**: YouTube's popularity surged significantly, particularly from 2010 onwards.
- **Significance**: The rapid growth of YouTube reflects the platform's evolution into a primary source of online video content.
- **Expectations**: The rise matches expectations, considering YouTube's dominance in the online video-sharing space.

Facebook:

- **Trend**: Facebook's popularity peaked in 2010 and gradually declined in the following years.
- **Significance**: The decline might be influenced by the rise of other social media platforms and changing user preferences.
- **Expectations**: Facebook has been one of the oldest platforms in existence, so it makes sense that it would have the highest influence at the earliest year. As history goes, Facebook has been known to have slowly fallen into decline in usage, so it makes sense that it would slowly do so in the following years. I am honestly unsure of what happened in 2010 to make Facebook peak so much.

LOL (Laughing Out Loud):

- **Trend**: The usage of "LOL" remained relatively stable over the years.
- **Significance**: "LOL" is a common internet acronym, often used in casual online communication.
- **Expectations**: Internet jargon and slang has always been revolving around all social media platforms, so it isn't too surprising that "LOL" remains as commonly used online everywhere.

:) :

This was another word/emoticon I was interested in, but it is not pictured in the graph as I realized that the counts were in the twenty thousands all throughout the years, which would make the graph completely unreadable due to the high numbers. I should have expected this though, as all emojis are extremely popular to use, even today.

Week 2 results

Analyze the total time it takes for your different data structures in the buildMap method of your WordCounter class year by year. I would average it over several runs per structure for each year to get more accurate results.

Time Taken (secs)	2008	2009	2010	2011	2012	2013	2014	2015
BST	35.068	83.774	107.073	89.897	89.607	74.747	89.519	86.292
HashMap	13.085	43.303	38.454	35.146	34.734	31.842	34.769	37.376

From the data, it is evident that the HashMap consistently outperforms the BSTMap in terms of the time taken to build the map for each year. HashMaps have an average runtime significantly lower than BSTMaps for all years, indicating that HashMaps are more efficient in handling the insertion of data, especially when dealing with large datasets. This efficiency is crucial for applications that involve frequent updates or modifications to the data structure. The results align with the expected behavior, as HashMaps generally offer better performance for insertion operations due to their constant-time average complexity for these operations.

Analyze the maxDepth() of your data structures after calling the buildMap method of your WordCounter class year by year.

maxDepth()	2008	2009	2010	2011	2012	2013	2014	2015
BST	225	145	99	110	485	2141	909	541
HashMap	9	10	9	8	11	9	10	10

From the data, it is evident that the BSTMap has significantly higher depths compared to the HashMap. This is because the structure of a BST depends on the order in which elements are inserted. If the elements are inserted in a sorted or nearly sorted order, it can lead to a skewed tree, resulting in a higher depth. On the other hand, HashMaps, when properly implemented, maintain a relatively balanced structure, leading to lower depths. The consistent lower depths of the HashMap across different years indicate that it is more resilient to variations in input data order compared to BSTMap. This makes HashMaps more efficient in terms of search, insertion, and deletion operations, especially when dealing with large datasets and varying input patterns.

Reflection

Why did we implement a HashMap in this project? How does it compare to the BSTMap of last week (we're looking for a summary of the results from the results/extensions sections here, basically).

The implementation of a HashMap in this project was crucial because it provides a more efficient way to handle word frequency data, especially when dealing with large datasets. In comparison to the BSTMap used in the previous week, HashMaps consistently outperformed BSTMaps in terms of the time taken to build the map for each year. HashMaps exhibited significantly lower average runtimes, indicating their superiority in handling insertion operations, especially with large datasets. This efficiency is vital for applications involving frequent updates or modifications to the data structure. Additionally, when analyzing the maxDepth() of the data structures after building the map, HashMaps consistently demonstrated lower depths compared to BSTMaps. This is significant because lower depths in data structures indicate a more balanced and efficient organization of data, leading to improved performance in search, insertion, and deletion operations. The consistent lower depths of HashMaps across different years highlight their resilience to variations in input data order, making them more efficient in handling diverse datasets. In summary, the implementation of HashMaps in this project was driven by their efficiency in handling large datasets, quick insertion times, and the ability to maintain a balanced structure even with varying input patterns. These characteristics make HashMaps a superior choice over BSTMaps for applications requiring efficient handling of word frequency data and similar datasets.

As a follow-up, if you use the entrySet method to write your wordCount files, on a non-self-balancing BSTMap you probably won't be able to read in a large file that was written by a BSTMap. Why is this?

When using the entrySet method to write word count files in a non-self-balancing BSTMap, the entries are written in the order they exist in the tree, which is determined by the insertion pattern. If the tree becomes unbalanced due to a skewed insertion pattern, such as inserting sorted or nearly sorted data, the tree can degenerate into a long, linear structure. In this scenario, the depth of the tree becomes equal to the number of elements, resulting in a time complexity of O(n) for insertion and retrieval operations. When reading a large file written by an unbalanced BSTMap, the time complexity for inserting each entry becomes O(n), making the process extremely slow and inefficient. This inefficiency arises from the lack of balancing mechanisms in non-self-balancing BSTs, causing their performance to degrade significantly when handling large datasets, especially during insertion and retrieval operations.

Extensions

SuperiorWordCounter

As stated earlier, there were errors with the process of counting words and surfing through the tens of thousands of different words used throughout the document, so I made a class SuperiorWordCounter to ignore those special cases. Special characters are removed from words using regular expressions (line.replaceAll("[^a-zA-Z]", "").toLowerCase()) before processing.

How do the most frequent words change year by year? Does it match your expectations?

Apart from the common words used to make sentences, such as “and”, “I”, and others which stay up consistently high, most frequent words come from whatever trends on social media during those years. One exception are emoticons as shown before are used constantly no matter what the year, as they are basically seen as a global language towards expressing oneself.

President Names

- One of the most frequent words was “Obama”, which makes sense as 2008 is the same year as the election Obama competed against McCain in. As he proceeded to win that election, the following years mention the specific acts and services Obama provided during his presidency, such as Obamacare. Eventually, people would mention his name less until his second more famous election run against Romney in 2012, having around 2,423 words. This makes sense as something as important as an election is bound to start arguments and discussion between people of varying political beliefs, to which Reddit is a perfect outlet for. I’d imagine that this would be the same for all presidential candidates.

Baby

- This is the same year Justin Beiber’s most infamous song “Baby” released to the public, having such an uproar of both positive and negative reception was surely to make itself known as discourse over reddit, as people will commonly use social media to clash opinions. As it shows, the results were no different. In that same year Baby was used 3,035 times, compared to the previous years where it was used for half that amount. However, it’s important to acknowledge that baby is a commonly used slang word, so without knowing in what context each use of the word is from I cannot say for sure that it came from the release of the song.

Minecraft

- In 2011, Minecraft was formally released to the public, and it became an immediate success, so it made sense that as the years following 2011 progressed, the mentions of Minecraft

would get more and more high in quantity. This also included aspects of the game, how to beat it, and creators who talked about the game.

Reddit

- Users often refer to the platform they are using by its name, especially in online discussions. On Reddit, discussions can be meta, involving the platform itself, such as discussions about features, subreddits, or general Reddit-related topics. Users might mention "Reddit" when talking about the platform or its community. Reddit is a content aggregator and discussion platform covering a vast array of topics. Many posts and comments discuss content found on Reddit, including trending posts, popular subreddits, or even mentions of other users' Reddit activity. This self-referential nature contributes to the word "Reddit" being frequently used within the platform. Reddit users often identify themselves as part of the Reddit community. They might discuss experiences unique to Reddit, such as inside jokes, common trends, or specific terminologies used within the community. This sense of belonging and shared identity leads to the word "Reddit" being a common topic of discussion.

Include 1 paragraph in the extension section of your report describing how the lab helped you complete the project, or how it could be improved to be more helpful.

Engaging in the lab sessions proved immensely beneficial in the completion of this project. The opportunity to collaborate closely with Teaching Assistants (TAs) and the professor within the same room greatly facilitated troubleshooting various challenges encountered during the project. Having immediate access to expert guidance not only saved valuable time but also provided valuable insights and alternative perspectives that significantly enhanced the project's quality. The interactive environment fostered a sense of collaboration, enabling us to discuss intricate problems, exchange ideas, and explore innovative solutions collectively. However, there is always room for improvement. A more structured approach, such as designated Q&A sessions or workshops focusing on specific project components, could further enhance the lab's effectiveness, ensuring that students receive targeted guidance in areas where they encounter the most challenges.

References/Acknowledgements

A lot of troubleshooting was done in order to make this work, the TAs Professor Lage and Professor Bender were both consulted as there were many, many errors. Although most errors occur because I missed very small things with my code.
