Analysis of Ranking Algorithms For the Use of Tweets Ranking

***Abstract*— Purpose of this research is to find the most optimized ranking algorithm which is capable of sorting tweets based on retweet and favorite count as well as the life times of those tweets. Our use case is to find the top 200 tweets which could be considered as most popular tweets at a given time for a particular candidate in an election.**

I. INTRODUCTION

For this research, expected behavior is that the rank of the tweets should be inversely proportional to the life time of tweets and directly proportional to the retweet and favorite count of those tweets. Within the scope of this research, we need to extract the tweets in the top 200 ranks so that latest news should appear fast and about 20% of the items should stay at the top in long term within that top 200 range.

II. PROCEDURE OF RESEARCH

A. Analyzing the initial data set within 24 hours period which is streamed by specific hash tags

First we collected the data which belongs to hash tags of different candidate within 24 hours. Since we would consider retweet to be of more worthy than a favorite because it is a double endorsement of opinion (agree + spread message to your own network) as opposed to just favorite (agree), we consider influence of retweet is double the value of favorite for popularity. From the 147490 large data set we collected, we could plot histogram of $2 \times Retweet + Favorite$ as indicated in figure 1.

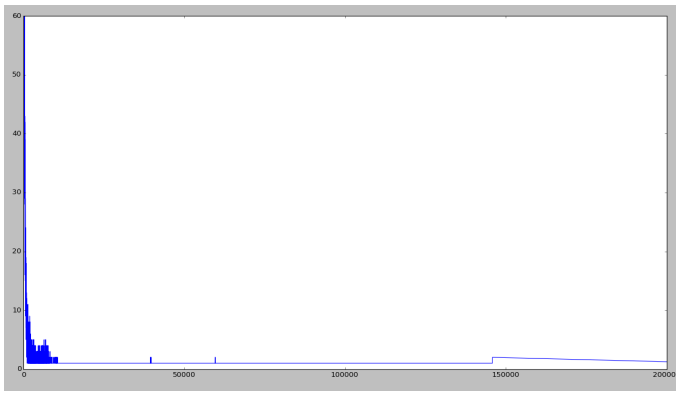


Fig. 1. $2 \times Retweet + Favorite$ count distribution.

(Life time distribution)

In the summery, data set contained the tweets which had maximum $Retweet + Favorite$ (not $2 \times Retweet + Favorite$) of 140380, maximum Retweet only is 78000. There are 10547 tweets which get $2 * Retweet + Favorite$ within a single day life time and maximum $2 \times Retweet + Favorite$ is 218582. 99 percentile of $2 \times Retweet + Favorite$ of that distribution is 39812. 92 and 99.9 Percentile is 218563.796.

B. Analyzing the algorithms

As the initial stage, we select 3 algorithms which are commonly used for ranking. They are

1) Reddit ranking algorithm:

$$ln(R) - \lambda t \tag{1}$$

2) Reddit Modified ranking algorithm:

$$ln(R) - \lambda t - ln(1 - e^{-\lambda t}) \tag{2}$$

3) Hacker News Formula:

$$\frac{votes - 1}{(item\_hour\_age + 2)^{gravity}} \tag{3}$$

We substitute $2 \times Retweet + Favorite$ as R and the life time of tweet as a t. Examining the minimum rank of top 200 tweets variation with retweet count vs life time of the tweet helps us to compare the the sensitivity(number of retweets and lifetime needed to enter the top 200 and rate of variation within top 200 with the retweet count) of 3 algorithms. Figure 2 shows the graphs which 3 colors represent the 3 algorithms (Blue-¿ Hacker News, Green(Red)-¿ Reddit 1 and 2) and variation in same algorithm with lambda value from 1 to 15 increasing by unit 1 at a time.

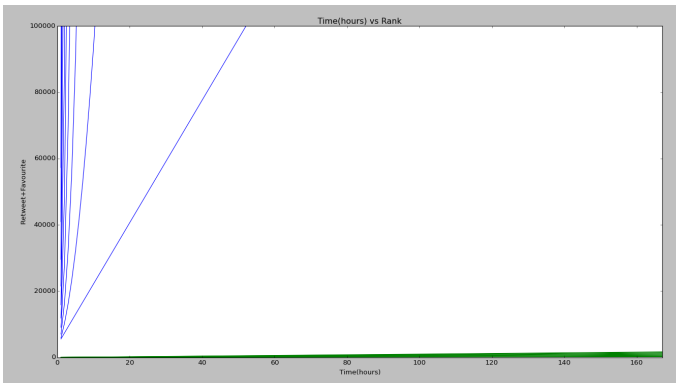


Fig. 2. Name the graph

(Blue-¿ Hacker News, Green(Red)-¿ Reddit 1 and 2)

From this graph we could identify that Hacker news formula is much sensitive for the retweet count since it shows an exponential behavior.

Then we plotted the behavior of lifetime of tweets which enter the window within 24 hours time (figure 4 and figure 5).

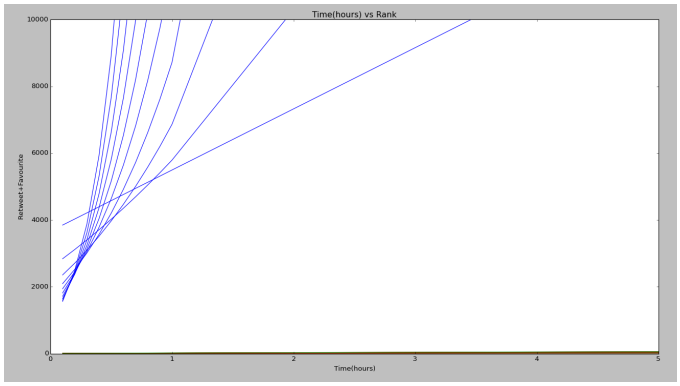


Fig. 3. Name the graph

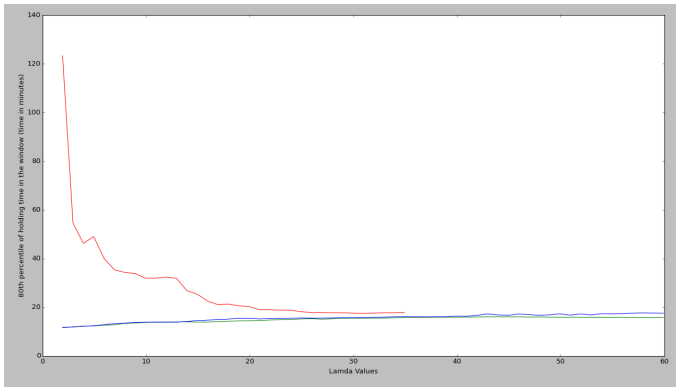


Fig. 4. Name the graph

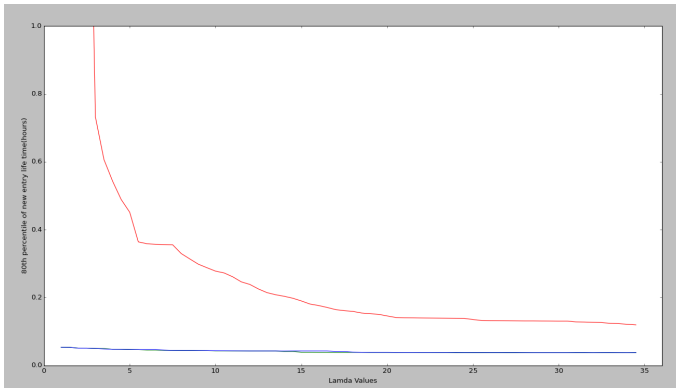


Fig. 5. Name the graph

(Red-¿ Hacker News, Green and Blue-¿ Reddit 1 and 2)

By this graph we can get some idea about the range of lambda value which we could select for our requirement. We need to enter the news item to the window as far as possible so the lifetime of the new entry tweets should be low. For that we could select lambda value greater than 10 for hacker news formula and any value for other 2 if we select those algorithms.

Next, we plotted the 80^{th} percentile of time spending inside the top 200 of the tweets which enter and exit from it within 24 hours. This is because we need the behavior such as latest news should appear fast and about 20% of items they should stay on the top long term. That is why we consider the 80^{th} percentile which we expected to get a large holding value.

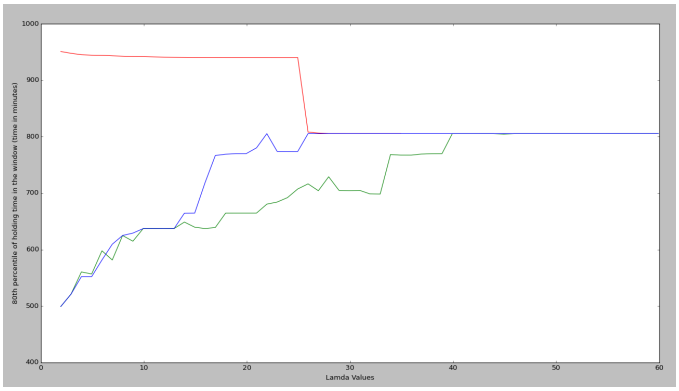


Fig. 6. Name the graph

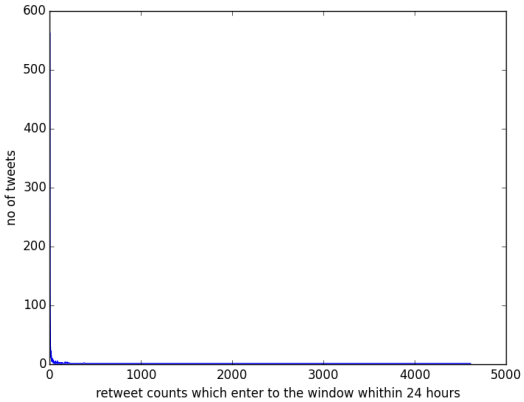
(Red-¿ Hacker News, Green and Blue-¿ Reddit 1 and 2)

By the graph in figure 6 we could get an upper limit of lambda value which we could use for hacker news formula. Looking at the 2 graphs, we could say that we can get some behavior which is closer to our expected behavior using lambda value in between 10-20.

C. Final test looking at top 200 window after 24 hours

Our next task is to find the most appropriate value for lambda between 10 and 20. For this, we consider all the tweets which enter the top 200 and consider the percentiles of those tweets starting from 99 up to 99.9 and select the lambda value which keeps the highest percentile value inside the top 200 window after the 24 hours.

Graphs in figure 7 and figure 8 shows the histograms of retweet and favorite count that have ended up in final top 200 window at the end of the day for lambda value of 11.



Graphs in the figure 9 and figure 10 show the Rank vs retweet and favorite count compared to the 99 percentile of tweets which enter the top 200 at the end of the day for lambda 11. Red dot displays the re tweet count of the percentile value.

When increasing the percentile values up to 99.8 only, lambda value of 11 for hacker news formula had kept that value within its' final top 200 window by end of the day. At

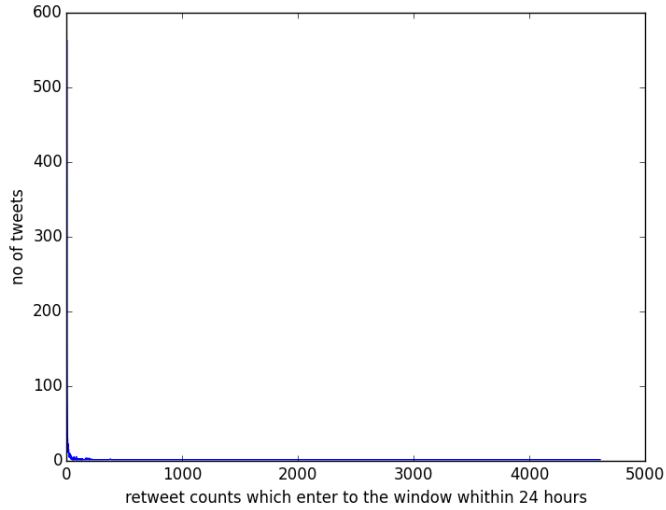


Fig. 7. Reddit 1 and 2

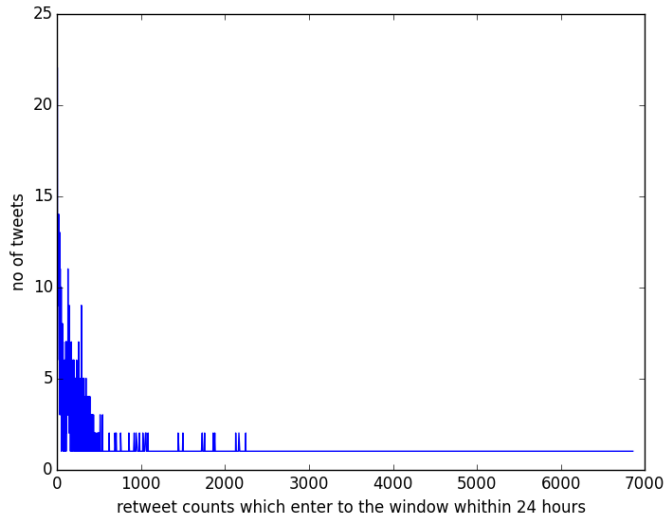


Fig. 8. Hacker News

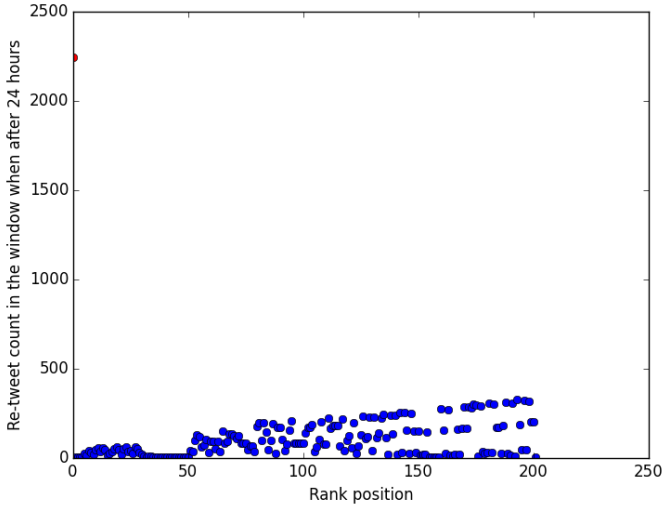


Fig. 9. Reddit 1 and 2

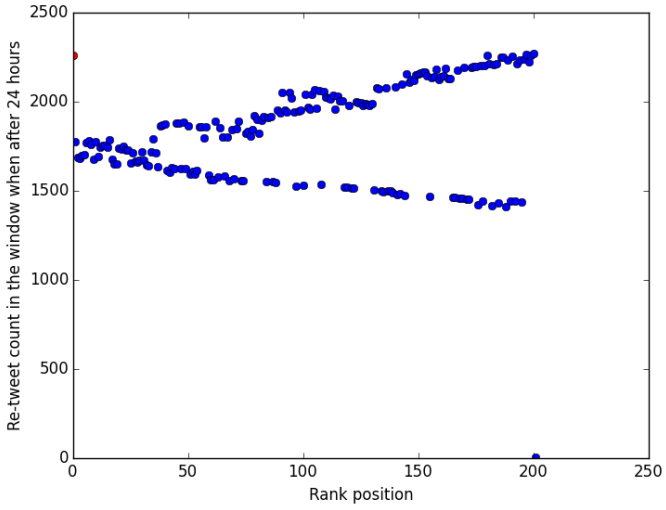


Fig. 10. Hacker News

that time, it had kept that tweet 8.919 hours and still it is at the top of the ranks.

By adding 1000 to all retweet + favorite count, we can uplift the holding time of 80^{th} percentile at the top 200 to 9.01 hours. At that time 80^{th} percentile tweet has 1420 $2 \times retweet + favorite$ count and 9.09472 hours lifetime.

By adding 5000 to all retweet + favorite count, we can uplift the holding time of 80^{th} percentile at the top 200 to 21.9 hours. But at that time 80^{th} percentile tweet has a too low retweet and favorite count which is equal to the 56. By adding 1 hour to all life times gives 8.9072 hours holding time within the window and $2 \times retweet + favorite$ count of 2358 while life time is 9.09472 hours.

III. CONCLUSIONS

By analyzing the above plots, we could say that by using hacker news formula, we can get a more live updating ranking system and by using lambda value of 11, we can increase the spending vale of 80^{th} percentile of each window. In addition, by adding 1000 to all $2 \times retweet + favorite$ counts, we can get a more reasonable behavior.