



# Tweet Selection

Let's see how the Tweet selection component fetches Tweets to display on a user's Twitter feed.

We'll cover the following



- Tweet selection schemes
  - New Tweets
  - New Tweets + unseen Tweets
    - Edge case: User returning after a while
  - Network Tweets + interest / popularity-based Tweets

## Tweet selection schemes#

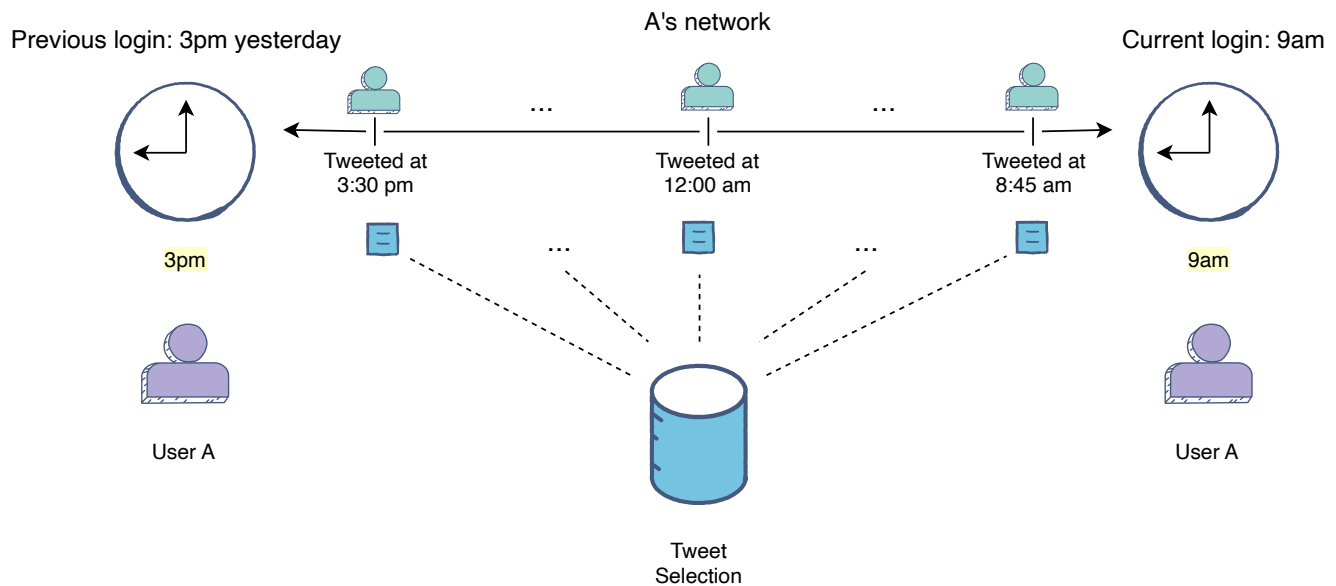
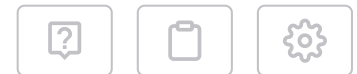
Let's see the various schemes Tweet selection component uses to fetch Tweets for a user's twitter feed.

### New Tweets#

Consider the following scenario to see how Tweet selection occurs.

User A logs in to Twitter at 9 am to view their Twitter feed. Now, the Tweet selection component has to fetch Tweets for display. It fetches the five-hundred newly generated Tweets by A's network since the last login at 3 pm yesterday.






Select new Tweets generated by A's network between 3pm to 9 am

### *A not-so-new Tweet!*

Consider a Tweet that user A has viewed previously. However, by the time the user logs in again, this Tweet has received a much bigger engagement and/or A's network has interacted with it. In this case, it once again becomes relevant to the user due to its recent engagements. Now, even though this Tweet isn't new, the Tweet selection component will select it again so that the user can view the recent engagements on this Tweet.

## New Tweets + unseen Tweets#

Picking up where we left off, let's see how this component may select a mix of new and old unseen Tweets.

The new Tweets fetched in the previous step are ranked and displayed on user A's feed. They only view the first two-hundred Tweets and log out. The ranked results are also stored in a cache. 

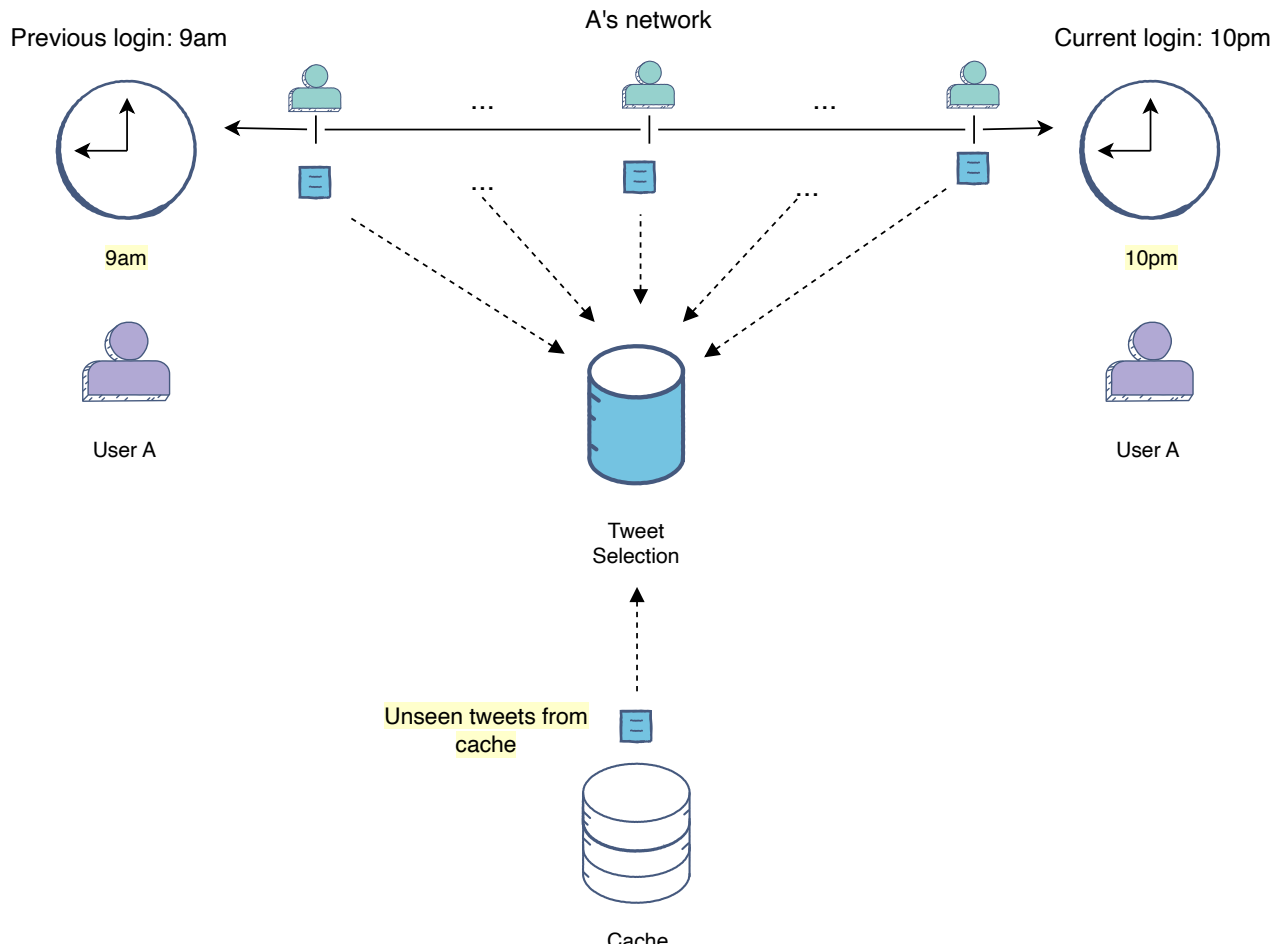


Now, user A logs in again at 10 pm. According to the last scheme, the Tweet selection component should select all the new Tweets generated between 9 am and 10 pm. However, this may not be the best idea!

For 9am's feed, the component previously selected a Tweet made at 8:45 am. Since this Tweet was recently posted, it did not have much user engagement at that time. Therefore, it was ranked at the 450<sup>th</sup> position in the feed. Now, remember that A logged out after only viewing the first two-hundred Tweets and this Tweet remained unread. Since the user's last visit, this unread Tweet gathered a lot of attention in the form of reshares, likes, and comments. The Tweet selection component should now reconsider this Tweet's selection. This time it would be ranked higher, and A will probably view it.

Keeping the above rationale in mind, the Tweet selection component now fetches a mix of newly generated Tweets along with a portion of unseen Tweets from the cache.



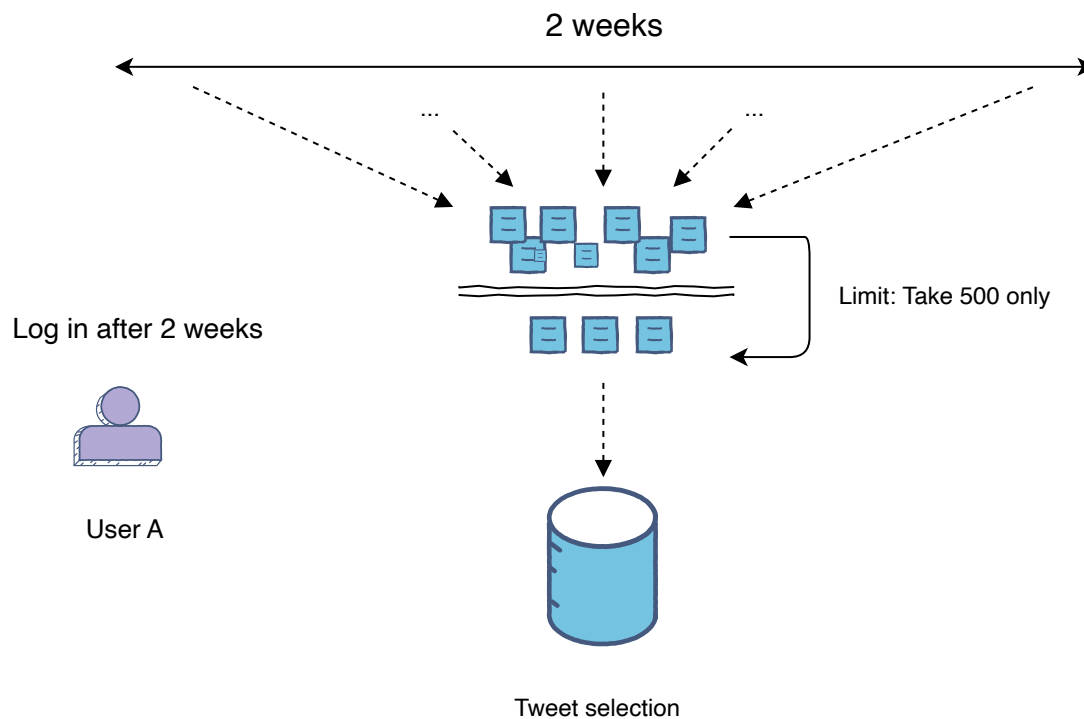
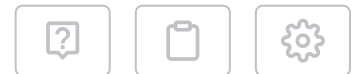


You may encounter the following edge case during Tweet selection.

## Edge case: User returning after a while#

Consider a scenario, where user *A* might log in after two weeks. A lot of Tweets would have been generated since *A*'s last login. Therefore, the Tweet selection component will impose a limit on the Tweet data it will select. Let's assume that the limit is five-hundred. Now the selected five-hundred Tweets may have been generated over the last two days or throughout the two weeks, depending on the user's network's activity. The main focus is that the pool of Tweets keeps on increasing so a limit needs to be imposed on the number of Tweets that the component will fetch.





Limit imposed on the number of tweets selected

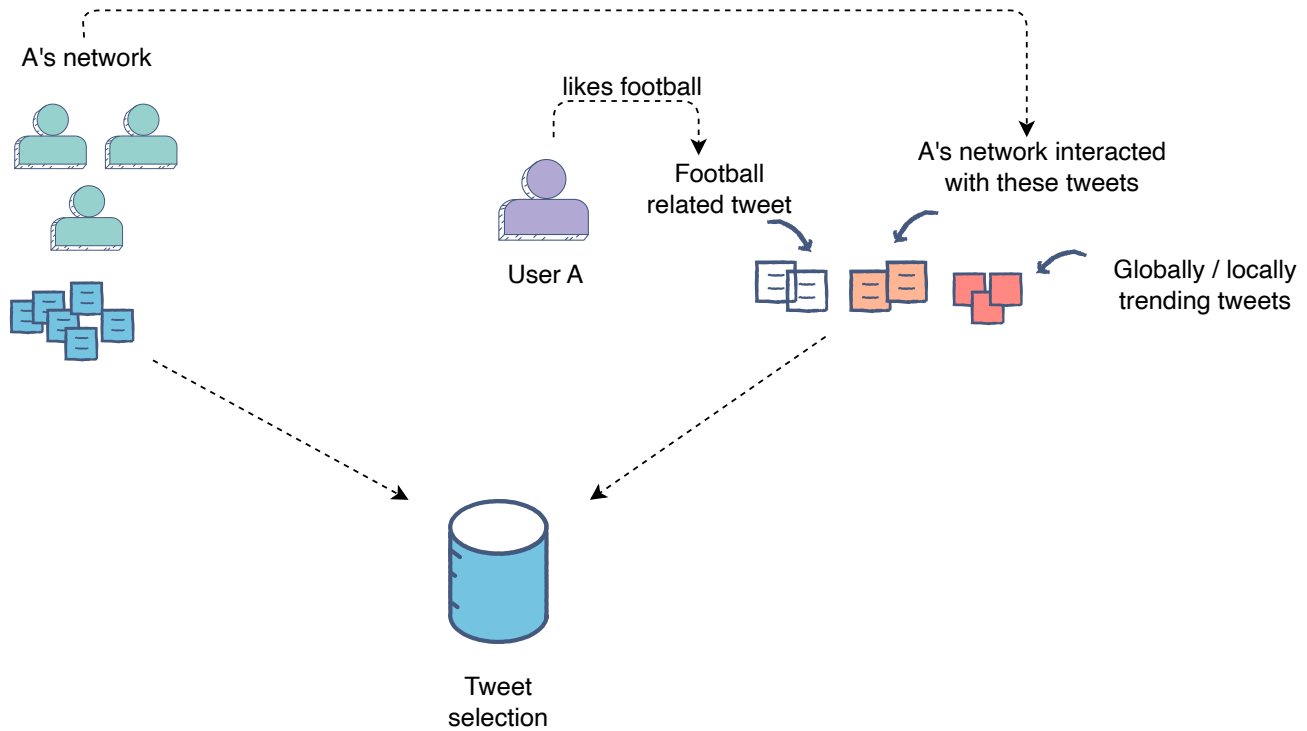
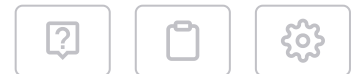
## Network Tweets + interest / popularity-based Tweets#

Until now, the Tweet selection component was only fetching Tweets from user A's network of followees. However, there could be Tweets **outside of user A's network** that have a high potential of engaging them. Hence, we arrive at a two-dimensional scheme of selecting network Tweets and potentially engaging Tweets.

An engaging Tweet could be one that:

- aligns with user A's interests
- is locally/globally trending
- engages user A's network





Selects network Tweets and interest / popularity-based Tweets

Selecting these Tweets can prove to be very beneficial in two cases:

1. The user has recently joined the platform and follows only a few others. As such, their small network is not able to generate a sufficient number of Tweets for the Tweet selection component (known as the Bootstrap problem).
2. The user likes a Tweet from a person outside of his network and decides to add them to their network. This would increase the discoverability on the platform and help grow the user's network.



All of the schemes we have discussed showcase the dynamic nature of the Tweet selection component's task. The Tweets it will select will continue to change each time.





Architectural Components



Feature Engineering

☒ Completed

 Report an Issue

