Scaling Tweet Reply Ranking to Tens of Million of QPS

Rishabh Misra

Machine Learning Engineer
Content Quality @ Twitter

3,008,107,408

Outline

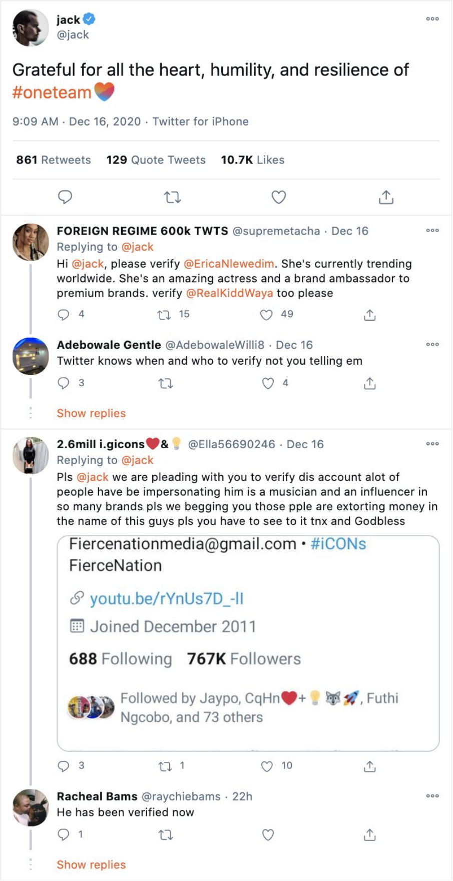
- Conversations product surface
- High-level overview of ranking pipeline
- Traffic nature and growth
- Scaling Approach
    - Measuring system load
    - Defining quality of replies
    - Identifying quantity of candidates to prune
    - Experimentation
- Key Results
- Q&A

Conversations Page

- When users click on any tweet on Twitter, they are taken to the *Conversation* page.

- On the page, users can see the conversation happening around the clicked tweet in form of nested replies.

Conversations on Twitter



jack
@jack

Grateful for all the heart, humility, and resilience of #oneteam🧡

9:09 AM · Dec 16, 2020 · Twitter for iPhone

861 Retweets **129** Quote Tweets **10.7K** Likes

FOREIGN REGIME 600k TWTS @supremetacha · Dec 16
Replying to @jack
Hi @jack, please verify @EricaNlewedim. She's currently trending worldwide. She's an amazing actress and a brand ambassador to premium brands. verify @RealKiddWaya too please

4 15 49

Adebowale Gentle @AdebowaleWilli8 · Dec 16
Twitter knows when and who to verify not you telling em

3 4

Show replies

2.6mill i.gicons❤️&💡 @Ella56690246 · Dec 16
Replying to @jack
Pls @jack we are pleading with you to verify dis account alot of people have be impersonating him is a musician and an influencer in so many brands pls we begging you those pple are extorting money in the name of this guys pls you have to see to it tnx and Godbless

Fiercenationmedia@gmail.com • #iCONs
FierceNation

youtu.be/rYnUs7D_-lI

Joined December 2011

688 Following **767K** Followers

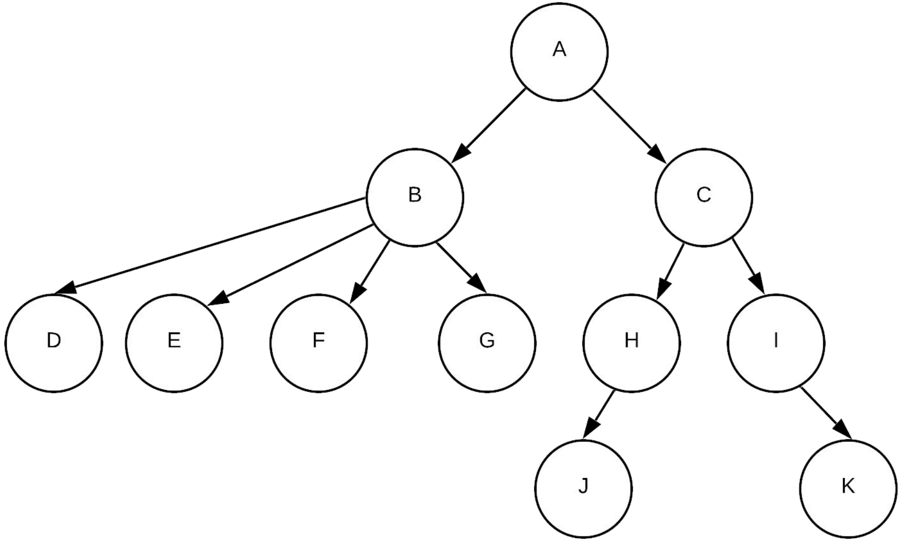
Followed by Jaypo, CqHn❤️+💡🐺🚀, Futhi Ngcobo, and 73 others

3 1 10

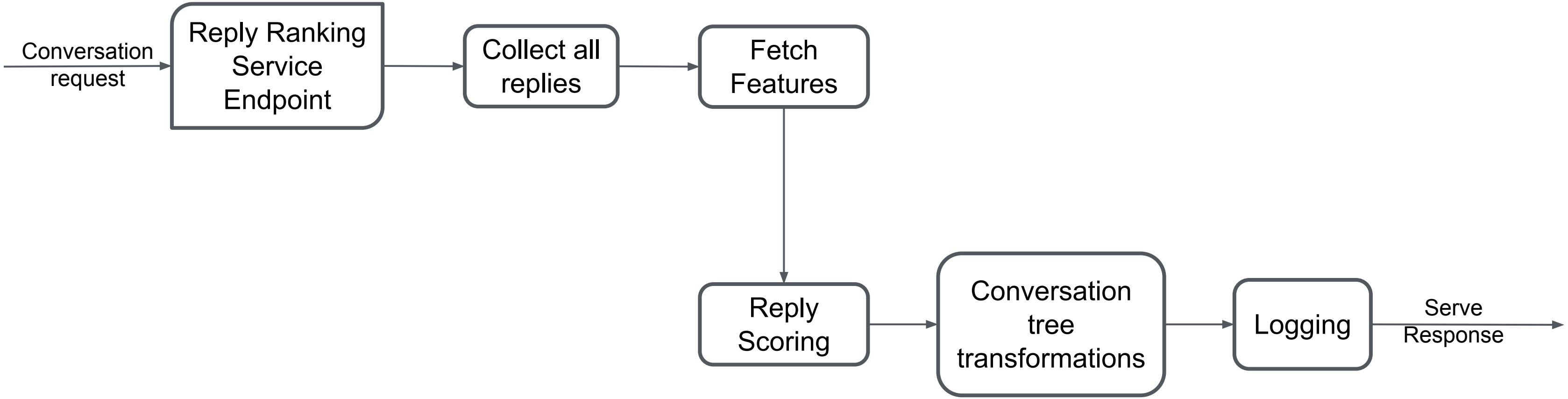
Racheal Bams @raychiebams · 22h
He has been verified now

1

Show replies

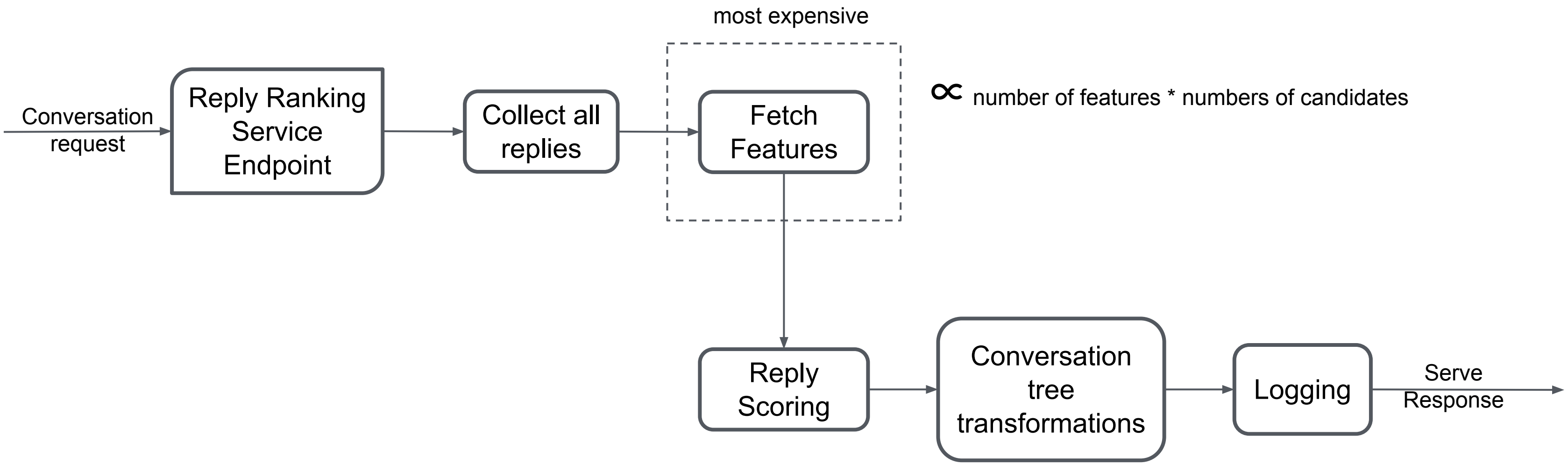


Pipeline Overview



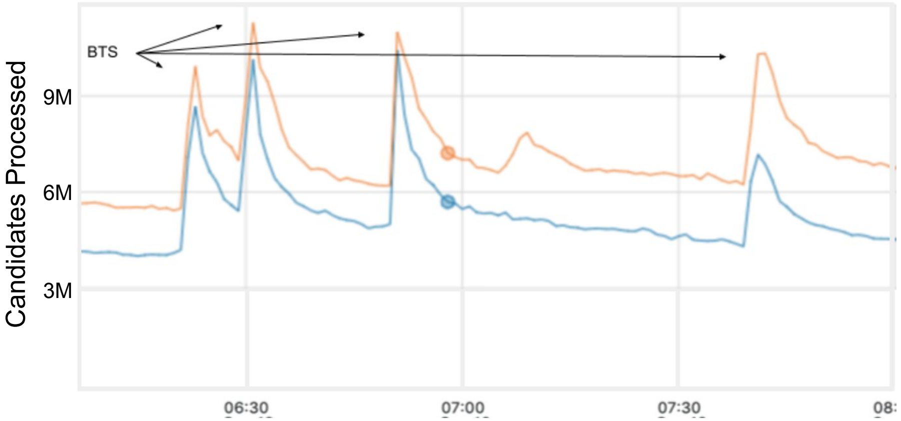
Goal of ranking models is to surface *engaging* & *healthy* replies that are *personalized* to viewers taste

Pipeline Overview



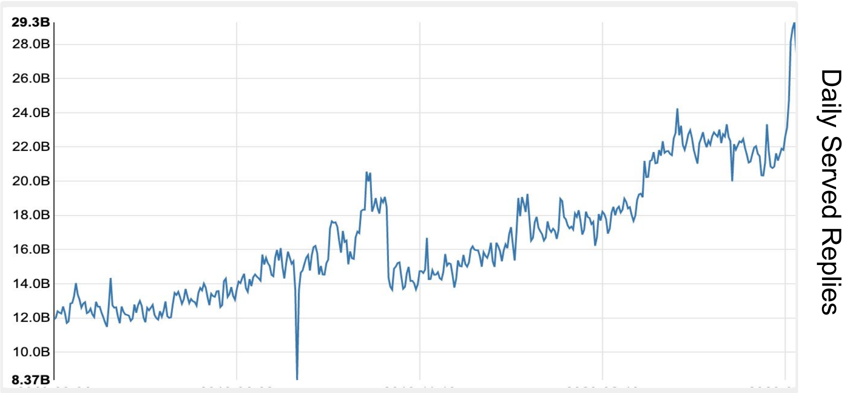
most expensive
Conversation request
Reply Ranking Service Endpoint
Collect all replies
Fetch Features
∝ number of features * numbers of candidates
Reply Scoring
Conversation tree transformations
Logging
Serve Response

Traffic Nature and Growth



When tweets goes viral (and external websites embed such tweets), the service experiences sharp increases in traffic.

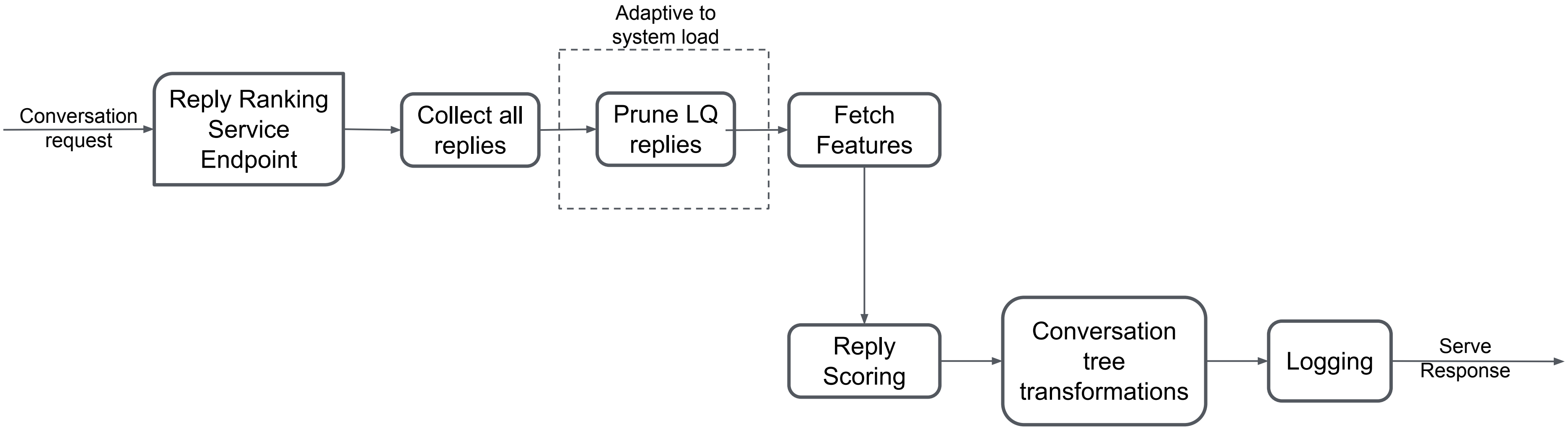
- Pictured is when BTS tweets went viral. Peak in recent times has been >20M.



Due to various product and ranking improvements, the service has been seeing organic growth in usage over the last few years.

- Pictured is served tweets more than doubled over 15 months period (12B -> 28B)

Scaling the Pipeline



Scaling the Pipeline

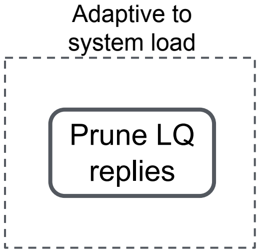
We thought about the problem from two perspectives:

- Computational perspective: reduce computations in the pipeline under higher system loads to make the service more robust.
- Business perspective: Make sure the scaling efforts keep the impact to key product metrics minimal to none.

We used computational perspective to shape the scaling strategy, and business perspective to refine the strategy.

Scaling the Pipeline

- Measure system load
- Define quality of replies early in the pipeline
- Calculate quantity of candidates to prune
- Iterate with experimentation



Adaptive to system load

Prune LQ replies

Measure System Load

Since our strategy would be tied to identifying system load, we should think about factors that convey service's health. Some ideas:

- Success Rate
- Throughput
- Latency

Measure System Load

Since our strategy would be tied to identifying system load, we should think about factors that convey service's health. Some ideas:

- Success Rate
- Throughput
- Latency (we selected p9999 latency to identify system load)

Define Quality

- Identify the signals that can be used early on in the pipeline to quickly identify candidates' quality. There are some considerations:

Define Quality

- Identify the signals that can be used early on in the pipeline to quickly identify candidates' quality. There are some considerations:
    - Feasibility of getting the signals early on in the pipeline.
        - Does pipeline requires restructuring?

Define Quality

- Identify the signals that can be used early on in the pipeline to quickly identify candidates' quality. There are some considerations:
    - Feasibility of getting the signals early on in the pipeline.
        - Does pipeline requires restructuring?
    - How many signals to consider?
        - Using intuition, restrict the initial set to a minimal.
        - Incrementally add signals as needed based on experimentation results.

Define Quality

- Identify the signals that can be used early on in the pipeline to quickly identify candidates' quality. There are some considerations:
  - Feasibility of getting the signals early on in the pipeline.
    - Does pipeline requires restructuring?
  - How many signals to consider?
    - Using intuition, restrict the initial set to a minimal.
    - Incrementally add signals as needed based on experimentation results.
  - Low latency way to compute quality based on the signals.
    - Model vs rule-based?
    - Evaluate speed and scope of improvement.

Define Quality

- Identify the signals that can be used early on in the pipeline to quickly identify candidates' quality. There are some considerations:
    - Feasibility of getting the signals early on in the pipeline.
        - Does pipeline requires restructuring?
    - How many signals to consider?
        - Using intuition, restrict the initial set to a minimal.
        - Incrementally add signals as needed based on experimentation results.
    - Low latency way to compute quality based on the signals.
        - Model vs rule-based?
        - Evaluate speed and scope of improvement.

- We included engagement-based (e.g. engagement counts), health-based (e.g. toxicity / report model scores), and tweet metadata-based (e.g. if tweet is written by viewer) signals to define the quality using rule-based approach.

Identify Quantity

- To identify quantity of candidates to prune, we thought about following aspects:
  - How much should pruning vary as system load increases?
    - Device a system load factor

Identify Quantity

- To identify quantity of candidates to prune, we thought about following aspects:
  - How much should pruning vary as system load increases?
    - Device a system load factor
  - Under a given system load, how much should pruning vary based on request size?
    - Device a request size factor

Identify Quantity

- To identify quantity of candidates to prune, we thought about following aspects:
  - How much should pruning vary as system load increases?
    - Device a system load factor
  - Under a given system load, how much should pruning vary based on request size?
    - Device a request size factor
  - Can pruning affect product experience in some cases?
    - Requests with small number of candidates (e.g < 60)

Identify Quantity

- To identify quantity of candidates to prune, we thought about following aspects:
  - How much should pruning vary as system load increases?
    - Device a system load factor
  - Under a given system load, how much should pruning vary based on request size?
    - Device a request size factor
  - Can pruning affect product experience in some cases?
    - Requests with small number of candidates (e.g < 60)
  - Should we prune if system is not under load?
    - Considerations:
      - Key product metrics should remain flat
      - Heavy ranking should still have sufficient candidates to have scope of improvements in future.

Experimentation

- Iterate on the strategy keeping business goal in mind - that is, user experience should not be affected during high system load.

Experimentation

- Iterate on the strategy keeping business goal in mind - that is, user experience should not be affected during high system load.

- Perform extensive A/B testing to iterate on how quality is defined:
  - Identify areas of product metrics losses.
  - Iterate on adding more quality signals and refining the rule-based heuristic.
    - Perform offline data analysis as needed.

Experimentation

- Iterate on the strategy keeping business goal in mind - that is, user experience should not be affected during high system load.

- Perform extensive A/B testing to iterate on how quality is defined:
  - Identify areas of product metrics losses.
  - Iterate on adding more quality signals and refining the rule-based heuristic.
    - Perform offline data analysis as needed.

- Perform A/B testing to iterate on identifying range of pruning:
  - Decide lower range based on how much faster we need our pipeline to be.
  - Upper range such that no effect on product metrics.
  - Then, graceful degradation mechanism varies the pruning range based on system load.

Experimentation

- Iterate on the strategy keeping business goal in mind - that is, user experience should not be affected during high system load.

- Perform extensive A/B testing to iterate on how quality is defined:
    - Identify areas of product metrics losses.
    - Iterate on adding more quality signals and refining the rule-based heuristic.
        - Perform offline data analysis as needed.

- Perform A/B testing to iterate on identifying range of pruning:
    - Decide lower range based on how much faster we need our pipeline to be.
    - Upper range such that no effect on product metrics.
    - Then, graceful degradation mechanism varies the pruning range based on system load.

- After production launch, run a holdback A/B experiment to continuously monitor the effect of scaling over longer term.

Key Results

- No impact on key product metrics.
- We could prune up to ~55% candidates early in the pipeline without impacting key product metrics. (Long term holdback is also flat after several months of launching)
- Pipeline’s p99* latency reduced by >15% and graceful degradation made it more robust under higher load.
- We implemented monitoring dashboard to track the pruning behavior and added relevant alerts.

Questions?

📨 rmisra@twitter.com
🐦 rishabh_misra_

Thank you.