Influencing Sentiment Against the Flu Shot on Twitter

Marcus DeMaster, Zenobia Liendo, Matt Swan and Dave Huber

Abstract

An overwhelming amount of information is shared every day across social media - a significant subset of which is either incorrect or generally misinformed. In an effort to test our ability to sway support against misinformation, we set out with a simple twitter bot and a target: those who share negative sentiment about the flu shot, or its effects, to see if we could convince them otherwise. Our experimental design was constructed such that we would reply with one of three types of response: a placebo or one of two treatments. Ultimately, while we were unable to change minds in the ways we had hoped, we were still able to learn some compelling things about engagement and influence.

1 Introduction

In the following paper, we will share our experimental design, our trials and tribulations, and, most importantly, our findings. First, we'll discuss our design. Please keep in mind that the design with which we started our experiment is not necessarily the one on which we ran the final experiment. However, if we were to begin with a description of the final design, we would have glossed over the reasoning and motivation for many of the changes we made during our first and second pilot studies.

After setting the scene, we will detail our initial tweet intake pipeline. We'll also describe both our first and second pilot studies, along with our learnings and adjustments. Finally, we'll explore the final experiment and consider our data, its analysis, and ultimate findings.

2 Experimental Design

The research question we were trying to answer was simple. Using only replies to targeted tweets, could we influence, and ultimately *limit*, the spread of misinformation in social media? Our experiment is necessary so that we can gauge the response of the individuals who are actually sharing the information directly. Without engaging them, we couldn't hope to understand how we may or may not be able to influence their behavior.

Our hypothesis was that we could reduce the amount of retweets and likes of a targeted tweet - that is, a tweet displaying negative sentiment towards the flu shot - by sharing information in conflict to this negative sentiment. We expected that we would be able to observe a decline in the retweets and likes of the original targeted tweet

because we expected that, once provided authoritative information to the contrary, people would reasonably be convinced not to proliferate the misinformation.

2.1 Two Treatments and a Placebo

Our treatment consisted of three different tweets, composed of two items. First, a single-line, generic lead-in (e.g. "Here is some useful information about flu prevention:"). Second, a link to the article we were using to convince our subjects. There were three treatment types: the placebo and two treatment tweets. We were careful to select articles that would show preview images as well, as we felt that was an impactful way to distinguish our placebo and treatments.

For our placebo tweet, we selected an article posted at Huffington Post, entitled, *"The Best Food And Drinks To Fight The Cold And Flu"*. The image was a bowl of soup. Our motivation was to select something that would not be viewed as taking a stance. This general wellness reply would be unlikely to be divisive.

For our first treatment tweet, we selected an article posted at the Centers for Disease Control (CDC), entitled, *"Everyone 6 Months of Age and Older Should Get a Flu Vaccine Every Season."* The preview image featured an healthy family of what appeared to be Hispanic descent. This treatment was meant to be truly oppositional - it even used the word *vaccine* instead of *shot*, which was sure to rile some feathers.

The second treatment took a different angle. We selected an article posted by the Public Broadcasting Service, entitled, *"These families lost kids to the flu. Now, they're fighting to prevent more deaths"*. The preview image featured presumably the mother and father of one of the children. They were obviously distressed. The motivation here was to see if a personal story would be more effective than the evidence-based rhetoric of the CDC.

2.2 Meet Tom Wilson

Tom (@tomwlsn31) was our initial twitter bot (we would later add Ted and Tim, as well). Tom was specifically crafted to deliver our replies. He was intended to have a friendly avatar and was a self-described "healthy living enthusiast". Tom's Twitter profile was carefully crafted to ensure no obvious predisposition. We did not want those who received the reply to be able to make any reasonable conclusions about Tom's motivation other than promoting healthy living. We even contracted a third-party to help seed his profile with a few hundred followers as to keep up appearances that he was not (literally) born yesterday.

2.3 Block Random Assignment

As we identified targeted tweets, it was important that we block their assignment based on two separate factors - number of followers and day.

First, the number of followers. This measure was sure to determine the reach of a particular Twitter user. Thus, treating a tweet from a user with a lower follower count compared to a larger follower count would have a potentially large impact.

Second, the day. Because of the volatile nature of social media, it was reasonable to assume that something might happen on a particular day which could skew sentiment. For example, if a particularly scathing article about the flu shot or vaccines went viral on a Monday, it may significantly impact users on that day, but be forgotten by Tuesday.

In order to properly block by these factors, while still maintaining random selection, we created a batch randomization process that initialized a random number - 0, 1, or 2 - that would correspond to each treatment, where 0 represented the placebo, for each block.

Let's walk through an example. An initial configuration could have randomly started as, [0,2,1,2,2]. When a tweet was evaluated, its block was determined. Assume our example tweet was in the third block. It would be assigned to 1 - the first treatment - and that value would be incremented, giving us [0,2,2,2,2]. In this way, we would ensure that we had an evenly distributed, randomly generated assignment, since we could not predict ahead of time how many tweets we might get in a day. Every day, the list we had been incrementing was dropped and a new list randomly generated.

This did lead to one complication, however. We needed a batch of tweets to get an accurate distribution by follower count to determine where our breakpoints should be. However, because we were replying in real-time, we needed to accept the assumption that the distribution would hold, as we could not set new follower count breakpoints constantly (and what would we do when we had 1 tweet? 3 tweets?)

2.4 Outcome Measures

Initially, we intended to measure the number of retweets and likes a post displaying negative sentiment for the flu shot received, observed both before and after treatment. The hypothesis was that - if our treatments worked - then we would see a significant slowing in how many times a tweet was retweeted or liked by other Twitter users.

We'll discuss our findings on this front in more detail during our analysis, but it is worth mentioning now that we additionally considered overall engagement with our

content, which included measures of how many times our content and profile link were clicked.

3 Building the Twitter Intake Pipeline

In order to complete our experiment, we were required to immerse ourselves in the Twitter API in two major capacities. First, we needed to construct a reliable stream that would be able to look for and collect all of the tweets in which we were interested in evaluating. Second, we needed to build a pipeline through which we could determine which tweets qualified as a subject, and which tweets should be treated, as well as implementing block random assignment and then, finally, posting the appropriate reply back to Twitter.

3.1 Streaming Flu Shot Tweets

Initially, we believed our pipeline architecture would need to be headlined by an industry-standard streaming application, such as Storm or Spark Streaming. However, once we got hands on with our target tweets, we realized that tweets including the search string, “flu shot”, come at a reasonable pace of about 40-50 per hour. As such, we did not need to implement a complex streaming architecture.

Our final design was a single python script, set to run as a service over an EC2 instance on AWS. While we initially streamed into an S3 bucket via Kinesis Firehose, we eventually switched over to put these tweets directly into a DynamoDB database. This script ran tweets into DynamoDB from November 13th until sometime on December 17th. In this time, it collected just over 30,000 total flu shot tweets.

3.2 Classifying Target Tweets

One of the key responsibilities of the intake pipeline was to classify which tweets we actually wanted to target. At this point, we had a streaming service that would grab all flu shot tweets, but we didn’t want to treat all tweets that came in. We wanted, specifically, those flu shot tweets that made claims that the shot was either ineffective or created some negative result (i.e. cause the flu or some other disease). We also had to overcome the challenge presented by the timing of the treatment. We were unable to classify the tweets by hand because we needed to decide, assign and treat in real-time.

In order to construct our classifier, we first needed to assemble a training dataset. We took the tweets that the pipeline was collecting and passed these tweets to Mechanical Turk for manual labeling. Using a custom template, we sent tweets 5-per-HIT (Human Intelligence Task). By sending multiple tweets in a single task, we could optimize the time it took to get results, as well as the overall cost. This cost

savings allowed us to send each set of 5 tweets to 3 turkers each and to use a majority vote system to determine which tweets truly exhibited the negative sentiment for which we were searching.

Using this method, we were able to amass a dataset of 9,000 labeled tweets, which we split 85%/15% into training and dev sets. Applying deep learning methods for natural language processing, we tested a series of different models to compare their performance. When testing on our dev set, we achieved ROC AUC performance scores of 83% with a logistic-regression-based model, 88% using a convolutional neural network (CNN) with embeddings, and 90% using a long short term memory (LSTM) model with attention and embeddings.

We chose to move forward with the LSTM with attention model, but adjusted our threshold from 0.5 to 0.8, in an effort to prioritize precision over recall. Largely, this worked, as the set of tweets we treated ended up greater than 60% true positives. In other words, a strong majority of the tweets we treated were the tweets we intended to treat.

3.3 Going Serverless

In order to keep maintenance time and effort low, we decided early on to implement a serverless architecture. Each step we needed to take as a result of the stream - classify, assign, reply - could be packaged up into a lambda function, to be called as needed. This was advantageous because we could focus more on properly implementing our experiment and refining our bots to avoid detection and, ultimately, muting or banning. We did not want to be spending this time or effort standing up or maintaining servers.

As previously mentioned, we were initially storing tweets from the stream into S3. This was a reliable, cheap method of storing, but would have required more effort to shape the data for further processing. We chose AWS Lambda for our functions because we learned that there exists a strong synergy between DynamoDB and Lambda. In short, any DynamoDB table can be transformed into a stream of its own, which can then be easily applied as a trigger for a Lambda function. Put more simply, any update to a streamed table would automatically run the lambda function on each record.

We immediately updated our stream to put entire tweets into a DynamoDB table. From here, it was straightforward process to build out the pipeline as we simply alternated between DynamoDB tables and Lambda functions.

3.4 Encapsulating the Classifier and Block Random Assignment

Using Lambda did require some upfront work to recreate our processes in the automated environment. Encapsulating the classifier was particularly tricky, mostly because of the overall file size and execution time limitations of Lambda.

The classifier itself was built using Keras, a machine learning framework that allows a programmer to use a single codebase with a few different machine learning backends. Initially, the classifier used Tensorflow for its neural networks, but Tensorflow proved to be too big to be able to run on Lambda. A clever solution, then, was to simply swap out the backend of Keras, switching from Tensorflow to Theano. Theano has a smaller footprint and allowed us to operate within the limitations of Lambda, while the use of Keras meant we could make this change without changing or rewriting a line of code.

We also ran into issues related to execution time on the classifier lambda. We had a maximum of 5 minutes in which to operate. Spinning up the environment took about 4 and a half. To combat this, we set the lambda to allow a higher memory allotment and also restricted the batch size to 1. Doing this meant the lambda would run more often, which meant the environment was more likely to stay up and not have to re-run every time the classifier did.

The block random assignment lambda offered significantly less resistance. As previously mentioned though, the original code was a batch process that randomly set a starting point and then handed out assignment sequentially. To convert this batch process into a distributed one, we simply created an additional DynamoDB table to store the current counts, so that as the batches came through, the process would remember where it was and preserve the proper balance in random assignments per block.

The reply lambda was the simplest at the outset of the experiment. All necessary information for the reply, including the block assignment, would be written to a table to be ingested by the reply lambda. It would then use the block assignment to collect the correct content and post the reply.

4 The First and Second Pilot

In the process of setting up for our final experiment, we were able to run two separate pilot experiments. We learned valuable lessons from both.

4.1 The First Pilot

Our first pilot study started just before Thanksgiving dinner and might not have made it through the entirety of Black Friday. The result was simple - we were suspended

by Twitter for a violation of their automation rules. Thankfully, a quick appeal of their decision was well-received and we were no longer suspended. We were warned, however, that if we were suspended again, it would likely be permanent. Unsure of how serious they were about that, we made changes to our execution plan to stave off further discipline.

4.2 Changes Made to the Pipeline

There were three crucial changes we made at this juncture. We added two more bots, for a total of 3. We added “filler” tweets to confuse the bot detectors. We modified our reply process to be less spammy.

This was the moment Ted and Tim were born. In an effort to distribute the automated replies, we created two additional bots, such that there would be a bot for each treatment. Tom delivered placebo replies, Tim delivered the CDC post, and Ted shared the personal story published on PBS News Hour. Everything else about these bots was the same, including their picture and background.

The second change we made was to add “filler” tweets. Essentially, we built a process to retweet from a handful of hashtags and user accounts that would reinforce our position on “healthy living”. If you viewed one of the profiles of one of the Wilsons Three, you’d find it reasonable that it could be a real person. The effects during testing were compelling as well… our test account gained dozens of followers while we were building the process, reinforcing the idea that we were on the right track.

It’s worth noting that the motivation for these first two changes was partly to the sanctity of the experiment. We feared that if a user clicked on our profile, they’d see all the treatments (replies you post show up on your public feed) and know we were a bot. We’d also experience spillover as a subject could now see all three treatments. This was resolved by our restricting one treatment per bot. The filler tweets acted in the same capacity, ensuring that if you clicked to one of our Wilsons’ you didn’t just see a long list of the same reply over and over.

Finally, we had to go back to the drawing board on the reply lambda. One of the reasons we felt we got suspended was because we were replying instantly to any tweet which we decided to treat. As a result, our new and improved delayed reply lambda did the following: it would check the current time against the time the tweet was assigned to a group. If the result was longer than 8 minutes ago, a reply would be sent to ensure we never replied beyond roughly 10 minutes. If it was anything less, a pythonic coin would be flipped. Heads… reply. Tails… back to the *treatedTweets* table. While we certainly replied to some tweets at the same instant, and some within a minute of being posted, our replies had been sufficiently staggered and our “Beware of Bot” sign wasn’t showing so obviously.

4.3 Additional Changes to the Experiment

While - technically - the following changes were indeed made to the pipeline, we consider them to be changes to the experiment itself that then manifested themselves through implementation.

Regardless, we added two checks prior to block random assignment. We decided that there was some recklessness to how we were choosing and treating tweets; specifically, we were not restricting user. It would have been possible for the same user to get multiple treatments if they had multiple qualifying tweets. This could be problematic with respect to spillover, so we added a check to ensure that if we had already replied to a specific user that we would not reply to them again.

The second check was with respect to a tweet thread. If someone posted something negative, and then someone else replied with an equally negative tweet, it was currently possible that we may have ended up replying to both! Whether we replied with the same or different treatments… we certainly would have disrupted things as we'd have clearly been advertising our bot status. Here we added a recursive check to get the root tweet id. If we had already replied to anyone in a particular thread, we would avoid replying there again.

4.4 The Second Pilot

Truth be told, we thought this second pilot was actually the final experiment… until Twitter got us again! This time, however, they did not suspend us. Clearly, we had built a pretty strong botnet; human-enough that Twitter wasn't willing to definitively suspend us. Instead, they muted us.

This turned out to be more damaging, however, because it happened quietly. During initial checks, all looked well. Just a few days later, however, things had changed. We expected to see our replies. In many cases, we saw the reply count as "1", but our reply still didn't show. After some digging, we realized that we had been muted. Essentially, Twitter hid all of our posts.

After some simple testing, we discovered it was because they were identical in content. Remember how we mentioned we were sending the same lead-in and link with every post? Well, apparently Twitter only allows you to send identical messages twice before flagging it for review. There obviously must be a time component to this as well, but we didn't hang around to discover every detail. We had enough information to make one final change to get things rolling again.

4.5 Writing Unique Lead-Ins

In testing, we found that if we changed some punctuation, the new tweet was still muted, but if we reasonably changed the lead-in, the tweet would be allowed to display. Begrudgingly, we split up the work and started writing unique lead-ins. We figured we needed to write around 65 unique lead-ins, assuming every duplicate could be posted twice, 65 unique lead-ins would give us about 130 replies for each bot, for a total of 390 replies before we'd be back in unfriendly territory. When you consider our roughly 60% true positive rate, that would give us enough observations for analysis.

By the end of it, we had written over 100 new lead-ins. We shuffled them into three distinct lists so that the bots were using different ones at different times and updated our delayed reply lambda one final time to grab a lead-in, pair it with the correct link and then reply via the correct bot.

4.6 Til the Bitter End

Despite our wonderfully hand-crafted lead-ins, we were not done doing battle with Twitter. We decided - given that when Twitter muted us before it literally went back in time and hid data from us regarding previously unmuted replies - that we would work as a team to login and download data roughly every 4 hours.

It turned out to be crucial to our efforts. We noticed as we were logging in to do data collection that each bot had - at one time or another during this final phase - its API credentials invalidated. This happened at different times, requiring us each time to generate new ones and update our lambdas. Thankfully, we were importing credentials via separate JSON files, so it was an easy update; annoying, nonetheless. In addition, they also required us to complete CAPTCHAs at times upon signing in to ensure we were human. Sometimes they even limited our abilities until we completed it, only then restoring full access.

Unfortunately, shortly after our presentation on December 14th, Twitter got wise and suspended "write access" for each of our accounts. We had our data and had begun analysis, so we yielded to the bot hunters and shut down the pipeline.

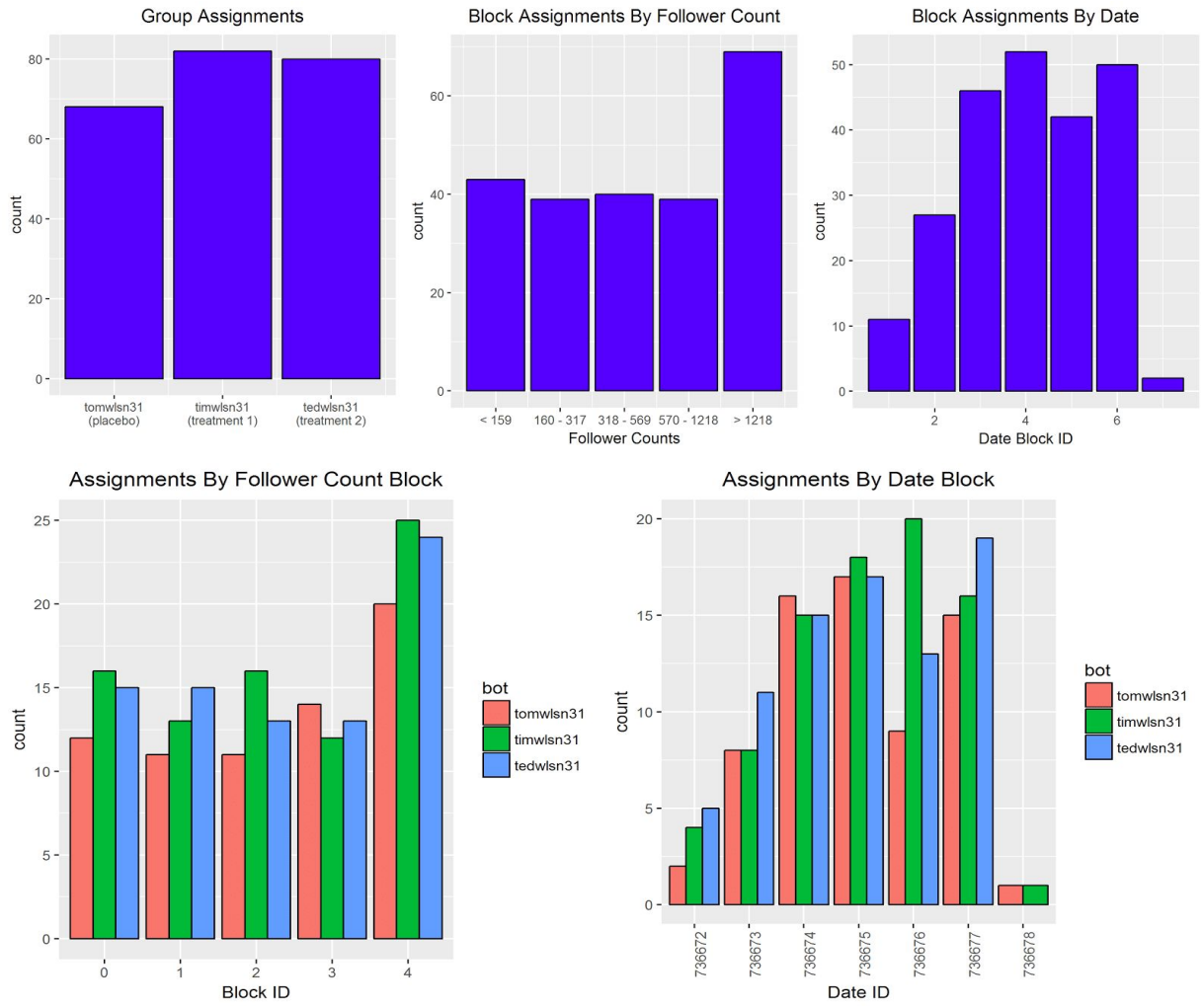
5 Data and Analysis of the Final Experiment

5.1 Group Assignment and Blocking

The three groups for the placebo, treatment 1, and treatment 2 were block random assigned according to follower count and the date of treatment. Figure 1 shows a summary of the final group and block assignments. The initial group assignments

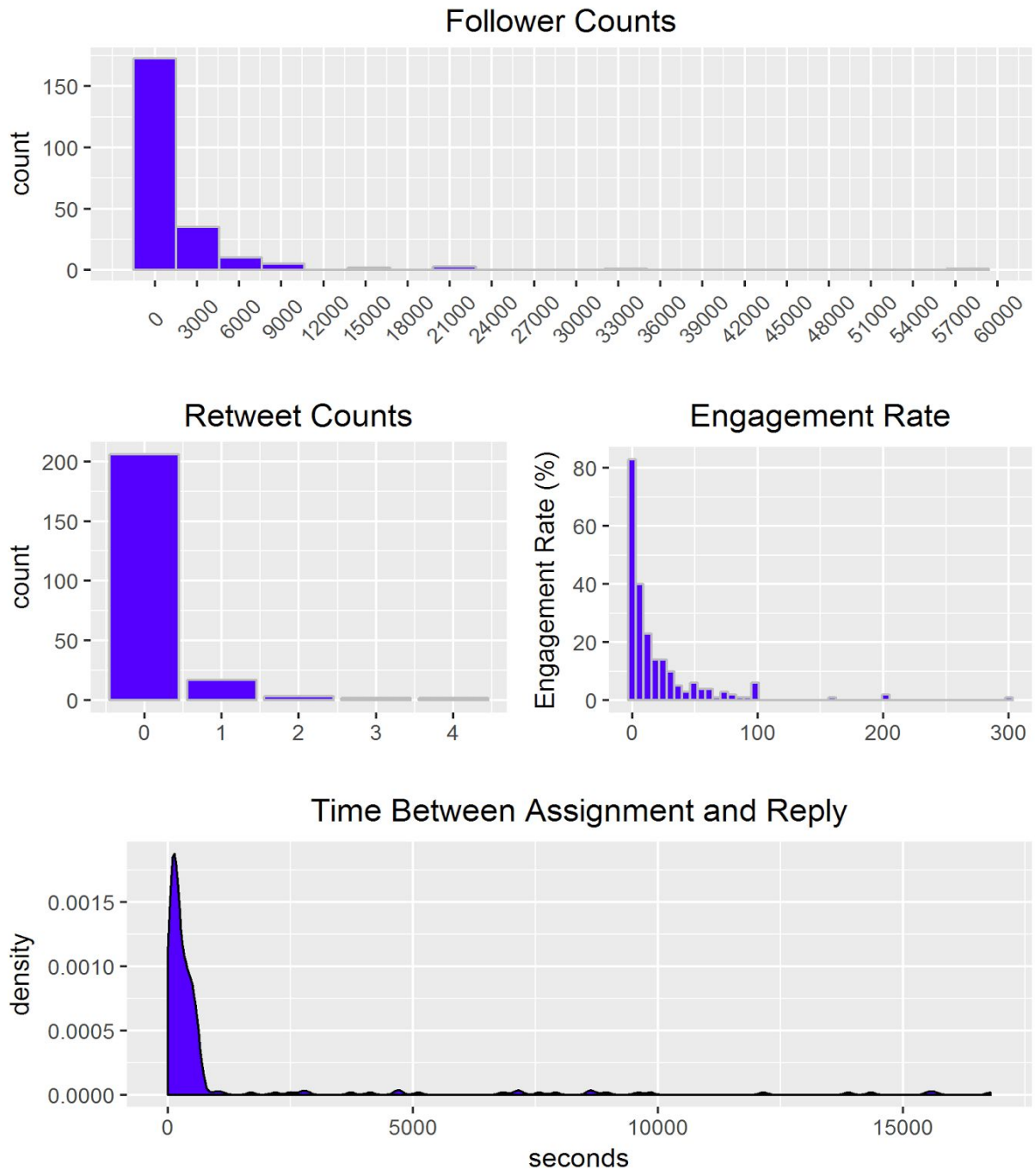
were an even split per group, but after removing undesirable tweets - classifier false positives - they have slightly different numbers. There does not seem to be any pattern to these tweets, so randomization is preserved. The highest follower count block exposed a discrepancy between the prior investigation of user follower counts and resulted in an overall block size of roughly 50% more subjects. The date blocks were expected to have variation in size. The first and last dates were not 24 hour periods and are much smaller, as expected. Except for one date that observed a noticeably higher assignment to treatment 1 (timwlsn31), they are all similar and show no group or block level patterns other than as noted previously.

Figure 1: **Group and Block Assignments**



(top left) The final size of each group at the time of analysis. *(top center)* The final sizes of the five follower count blocks. *(top right)* The final sizes of the date blocks. The first and last dates were not 24 hour periods. *(bottom left)* Composition of each follower count block. *(bottom right)* Composition of each date block.

Figure 2: Collected Data



(top) Follower counts of original tweet user. *(center left)* Retweet counts of the original tweet. *(center right)* Engagement with the experimental reply tweet. *(bottom)* Time between when the user was assigned to a group and block and when the reply was actually sent.

Figure 2 shows the data that was collected. The follower count distribution has an expected long tail, but contains more mass than expected. The number of retweets of the original tweet was quite low; however, the engagement rate on our reply is fairly active. The rate of engagement is calculated as follows:

$$Engagement\ Rate\ =\ \frac{number\ of\ engagements}{number\ of\ impressions}\ x\ 100\%$$

Impressions are when a tweet is seen by a Twitter user. Engagements are tweet interactions such as link clicks, profile clicks, likes, etc. The engagement rate can be higher than 100% because of the many types of engagements that a single tweet may undergo. A tweet is assigned in real time as soon as it streams into the pipeline. The random delay mechanism described above causes the time between assignment and reply to vary. Other factors at play here involve the occasional blocking and credentials invalidation. There was considerable variation here, so this measure was included for use as a covariate.

5.2 Models

For each outcome, retweet count and engagement rate, 3 models were built: a simple model, a blocked model, and a blocked model with one covariate. In both cases, the covariate used was the time between group assignment and reply tweet. Table 1 shows a statistically significant effect at the 5% level for treatment reply 2 with a P value of 0.03308 (calculation not shown). Table 2 shows a statistically significant effect at the 5% level for both treatment reply 1 and treatment reply 2 with P values of 0.0101 and 0.02921, respectively (calculation not shown). The tables report confidence intervals for these findings along with the average treatment effect sizes.

Table 1

Modeling Retweet Count

	Retweet Count		
	simple	blocked	w/ covariate
constant	0.294***	0.194	0.208
	(0.104, 0.484)	(−0.105, 0.494)	(−0.093, 0.508)
Treatment 1	−0.160	−0.158	−0.160
	(−0.381, 0.061)	(−0.396, 0.080)	(−0.400, 0.079)
Treatment 2	−0.219**	−0.223**	−0.225**
	(−0.421, −0.017)	(−0.428, −0.019)	(−0.431, −0.019)
Observations	230	230	230
R^2	0.025	0.071	0.072
Adjusted R^2	0.017	0.020	0.016
Residual Std. Error	0.560 (df = 227)	0.559 (df = 217)	0.560 (df = 216)
F Statistic	2.959* (df = 2; 227)	1.388 (df = 12; 217)	1.292 (df = 13; 216)
Note:			*p<0.1; **p<0.05; ***p<0.01

Note: The covariate and block level effects were removed for clarity. They were not significant. Confidence intervals are provided using robust standard errors.

Table 2

Modeling Engagement Rate

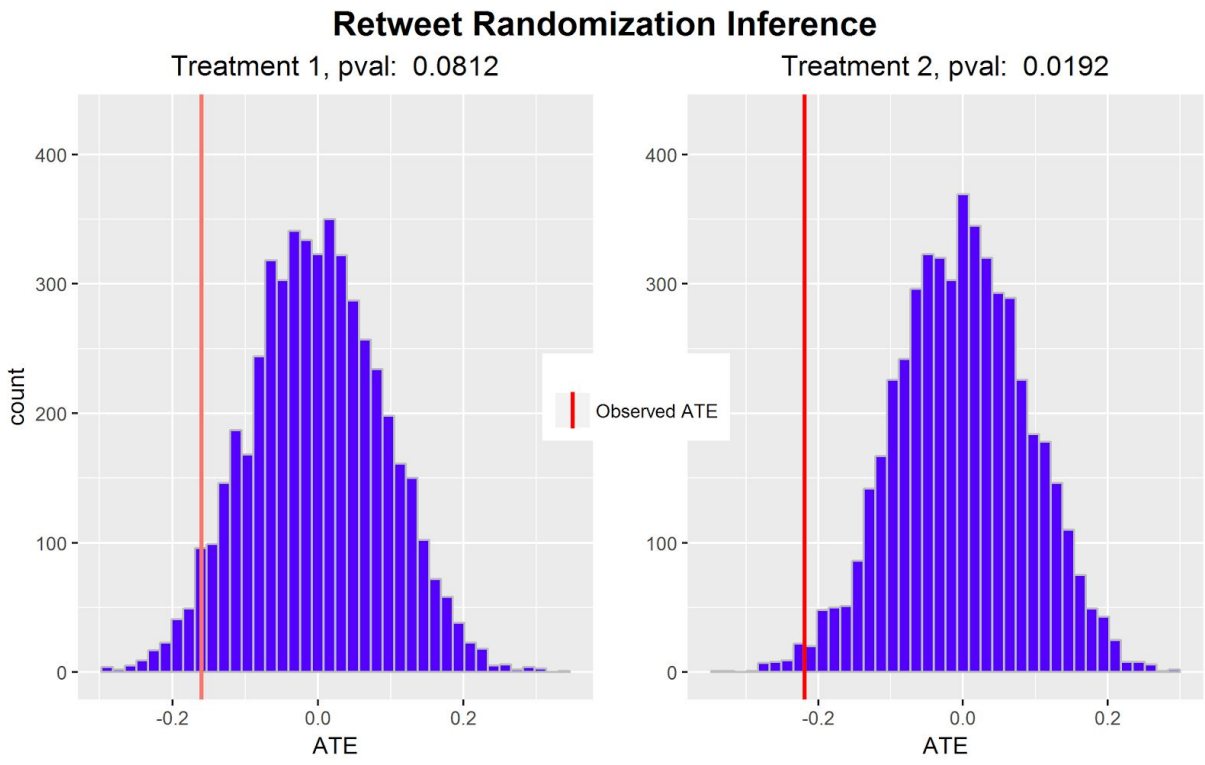
	Engagement Rate		
	simple	blocked	w/ covariate
constant	0.350***	0.460	0.437
	(0.210, 0.489)	(−0.180, 1.101)	(−0.167, 1.041)
Treatment 1	−0.212***	−0.214***	−0.209***
	(−0.358, −0.065)	(−0.375, −0.052)	(−0.366, −0.051)
Treatment 2	−0.189**	−0.190**	−0.187**
	(−0.338, −0.041)	(−0.360, −0.021)	(−0.353, −0.020)
Observations	224	224	224
R^2	0.064	0.080	0.086
Adjusted R^2	0.055	0.028	0.029
Residual Std. Error	0.351 (df = 221)	0.356 (df = 211)	0.356 (df = 210)
F Statistic	7.513*** (df = 2; 221)	1.527 (df = 12; 211)	1.513 (df = 13; 210)
Note:			*p<0.1; **p<0.05; ***p<0.01

Note: The covariate and block level effects were removed for clarity. They were not significant. Confidence intervals are provided using robust standard errors.

5.3 Randomization Inference

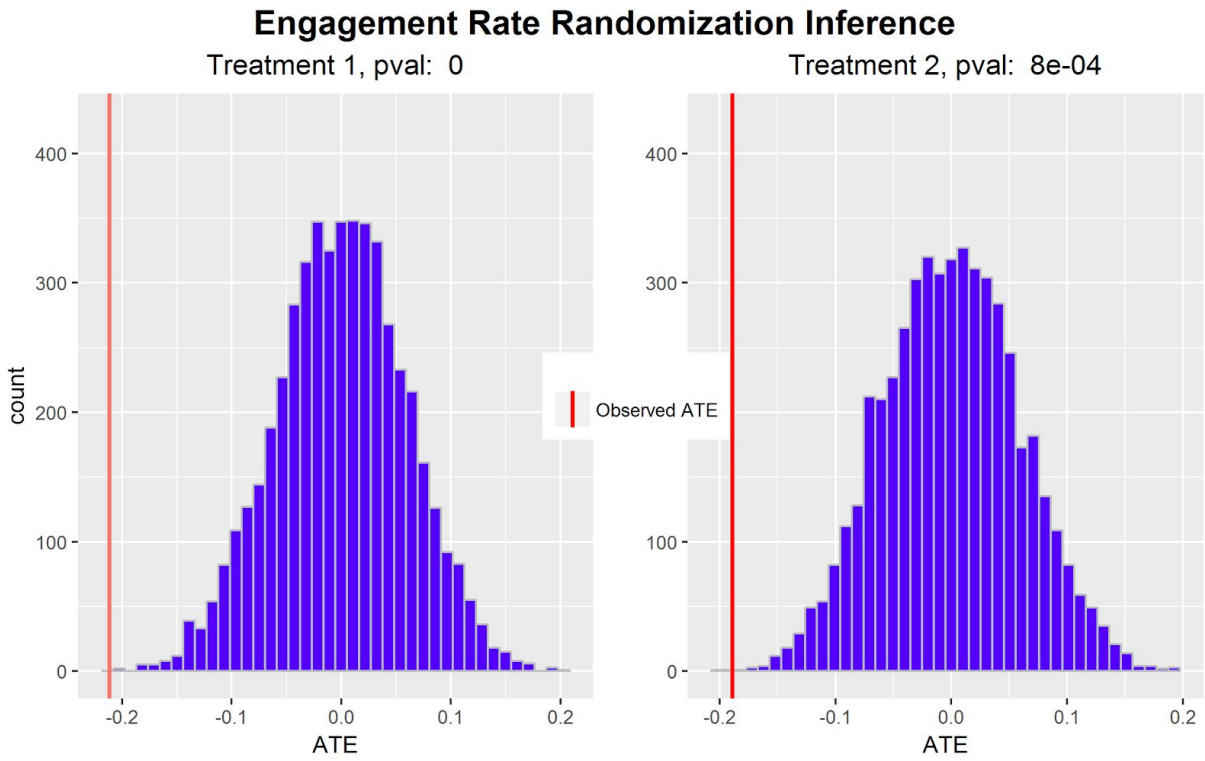
Randomization inference was used for both outcome measures to determine if we were merely witnessing a side effect of the particular randomization used in the experiment. Figures 3 and 4 show the results. In all cases, it seems unlikely that the observed effects were due to chance. In addition, it seems extremely unlikely that the retweet effect for treatment 2 and the engagement rate effects for treatments 1 and 2 were due to chance.

Figure 3: Retweet Count



(left) Randomization inference for the retweet count outcome for treatment reply 1
(right) Randomization inference for the retweet count outcome for treatment reply 2.
Note: The randomization was performed with 5000 simulations with randomized group assignment.

Figure 4: Engagement Rate



(left) Randomization inference for engagement rate outcome for treatment reply 1
(right) Randomization inference for engagement rate outcome for treatment reply 2.
Note: The randomization was performed with 5000 simulations with randomized group assignment.

5.4 Non-compliance

An investigation into non-compliance did not take place until after the primary analysis because it was not deemed necessary ex ante. It was later decided that tweets where our reply had no impressions and were identified as true positives were in fact cases of non-compliance. If it got zero impressions, it was never seen by anyone, and thus was not delivered. Since we are using a placebo group in addition to the treatment groups we can remove non-compliers from all groups and rerun the regression analysis. Table 3 shows the results of this regression with only compliers in the treatment and control groups. It can be seen that after removing non-compliers, precision is increased for both outcome measures across both treatment groups. Also, both treatments have an increased effect size for retweet count, although treatment 1 is still not statistically significant.

Table 3: Non-compliance

Compliers Retweet Count

	Retweet Count w/ covariate
constant	0.226 (−0.086, 0.538)
Treatment 1	−0.183 (−0.437, 0.072)
Treatment 2	−0.248** (−0.466, −0.029)
Observations	224
R^2	0.079
Adjusted R^2	0.022
Residual Std. Error	0.566 (df = 210)
F Statistic	1.378 (df = 13; 210)
Note:	*p<0.1; **p<0.05; ***p<0.01

Compliers Engagement Rate

	Engagement Rate w/ covariate
constant	0.437 (−0.167, 1.041)
Treatment 1	−0.209*** (−0.366, −0.051)
Treatment 2	−0.187** (−0.353, −0.020)
Observations	224
R^2	0.086
Adjusted R^2	0.029
Residual Std. Error	0.356 (df = 210)
F Statistic	1.513 (df = 13; 210)
Note:	*p<0.1; **p<0.05; ***p<0.01

(top) Retweet count model corrected for noncompliance. *(bottom)* Engagement rate model corrected for noncompliance. *Note: Provided confidence intervals are calculated with robust standard errors.*

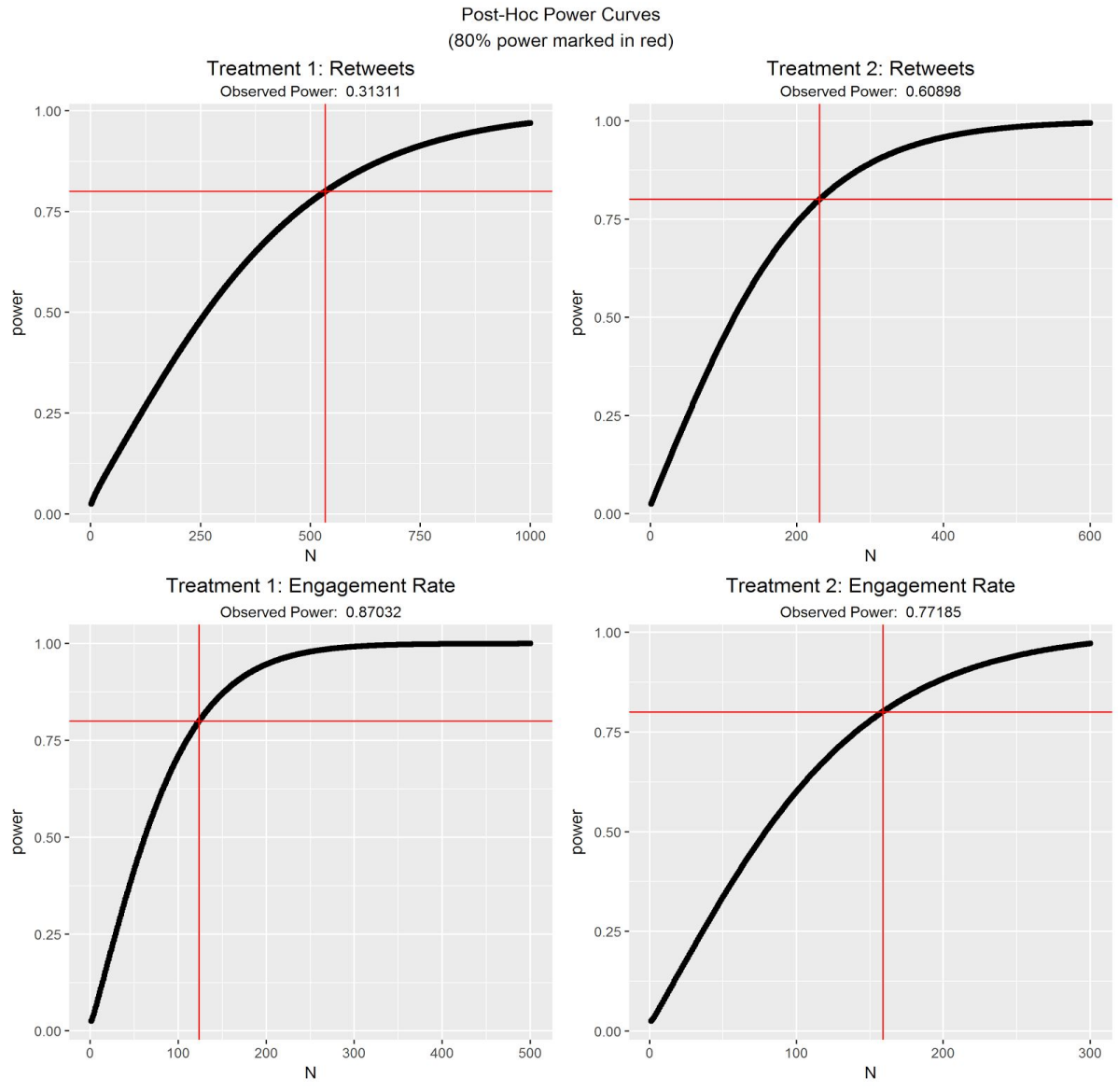
5.5 Attrition

There were 16 tweets that had been deleted or muted before data collection and are considered to have attrited. An analysis of the missing tweets determined that the attrition was not differential in nature and can be ignored.

5.6 Power Analysis

Without a clear idea of the effect sizes we were looking for, an ex ante power analysis was inconclusive. To aid in future experimentation, a post-hoc power analysis is provided in Figure 5 and can inform the necessary sample size to observe a consistent effect similar in size to what was observed here.

Figure 5



The red lines cross at the point where 80% power would be achieved assuming the observed distribution persists after more data collection. The power observed in this experiment is reported in the subtitle of each subplot.

6 Findings

6.1 Generating Influence

Ultimately, since the measure of action that we do have does not have a direct impact on public health - we don't have a measure, for instance, of who did and did not actually get the flu shot - we cannot make any claims about our ability to influence reasonably better health outcomes. However, what we got instead is still compelling.

Our initial intention was to reduce the number of retweets of the original tweet with negative perspective of the flu shot, where the content of our reply ran in opposition to the tweets themselves. If users were to see our content, it should offer an alternative perspective that we hoped would stop the spreading of this misinformation. In both cases, we saw an effect in the direction of reducing the number of retweets, but this effect was only significant with the second treatment - the story about the children dying from the flu shot. The effect of the CDC reply was not statistically significant.

This is compelling, however, if not also bittersweet. What it tells us about the world - of flu shot tweets - is that we can be successful in repressing retweets and limiting the spread of misinformation, given optimized messaging. In this case, that took the form of an emotional anecdote about the worst case scenario.

Let's take this a step further. Engagement, in a general sense, is a measure of people's willingness to be exposed to particular content. Their engagement with the non-threatening content is expected. They were ailing and we offered a reprieve without challenging any of their positions. We received many thanks and likes for this content.

On the other end, both treatments showed a negative effect with regards to engagement, in a statistically significant way. This, too, would be expected. By achieving a reduction in engagement, we find that not only have we stopped the spread of the endangering conversation; we've quieted the conversation itself.

We should be overjoyed. We won. However, we mentioned a bittersweet feeling. What gives us this feeling is the ground-shaking realization that our data and our findings cannot make one simple distinction. That is, have we *overcome*, or merely *suppressed* the misinformation and fake news? In other words, although we were successful in limiting the spread of this ideological epidemic, are we only treating the symptoms, and not the disease? Guess we'll need another experiment.

Appendix A - Lead-in Phrases

The following list of lead-in phrases was shuffled for each group assignment bot and cycled through to introduce the link in our reply tweets.

```
about preventing the flu:
Beating the flu
Fighting Influenza:
Fighting flu germs
Flu germs go away!  some information on prevention:
Flu info and prevention
Flu Information and Prevention:
Flu prevention information:
Flu Season survival info:
Getting the flu is horrible!  how to prevent it?
How to avoid catching the flu:
how to avoid spreading the flu
"Influenza, how to protect yourself and others: "
Info about protecting ourselves from the flu:
Info for Flu Prevention
Info for not spreading the Flu:
"Nobody wants to get the flu, this link may help"
"Preventing the flu, some info:"
Protect yourself from the flu:
Say no to Influenza!:
"Seasonal flu prevention, this link may be useful"
some more information about flu prevention
Stop the spread of flu
stopping the flu:
Stopping the spread of flue germs
Stopping the spread of the flu:
Surviving the Flu Season:
"the flu is contagious, some prevention information:"
Things to do to prevent the flu:
this information about flu prevention may be useful:
I was really surprised when I read this article about the flu:
This article about the flu really opened my eyes:
My eyes opened a little wider after reading this article about the
flu:
"Hey, when I read this article about the flu, I was pretty
surprised:"
Here is some info about the flu that might surprise you
```

Someone sent this to me earlier today and I can't stop thinking about it:
Somone shared this with me and I'm surprised how much I agree with it:
I want to share this article about the flu with you:
Everyone around us benefits from flu prevention:
Everyone plays a role in flu prevention:
We all share the responsibility of preventing the flu.
Preventing the flu is everyone's responsibility:
We can all do something to prevent the flu from spreading.
There's something we can all do to stop the flu.
Learning about the flu is as important as preventing it.
There is a lot of information out there about the flu. This cuts to the core:
This article about preventing the flu really cuts to the core:
Simple information about flu prevention:
I think you might find this interesting:
This might be interesting to you:
Maybe you'd be interested in this info about the flu:
You may want to check this out:
You might want to read this:
You might enjoy reading this:
Take a look at this info about the flu:
Here is some info about how to take action against the flu:
Here is what you need to know about flu prevention:
The flu is a problem that we can solve.
Have a look at this article about preventing the flu:
Can you take a look at this for me?
Can you look at this flu prevention article for me?
Can you read this article about flu prevention?
Would you be able to look at this article?
Have you read this article yet?
Helping to stop the spread of influenza should be an informed decision.
The decision to prevent the flu may be the most important one you make this year.
How you help prevent the flu should be an informed decision.
I think I've found something that would interest you.
This article may interest you.
You may find this information useful.
I think you could find this interesting
Maybe this information about flu prevention would be useful to you.
This article about preventing the flu may be useful.

Taking steps to prevent the flu is important.
There are lots of reasons why its important to take steps to prevent the flu.
I feel like you would find this interesting.
Maybe you would find this interesting.
This is good information I think.
You might find reading this helpful.
I found this information to be useful. Maybe you would too.
This information is useful imo. Maybe you'll find it helpful too.
Have you read this article? You may find it helpful.
This is an article I think you would find interesting.
Here's an article you might think is helpful.
This article is really informative. Have you read it?
You would find this information useful I think.
You will find this article interesting I think.
I'm guessing this article will be useful information for you.
"If you haven't read this, I recommend it."
I recommend reading this. It's very informative.
This article is super useful. You might like it.
This a really great article about flu prevention. Very informative.
We can all take steps to prevent the flu. This is some good information.
I found the article here to be useful.
"The flu is scary, but I found these tips helpful."
Ugh. I hate the flu. I hate getting sick.
Fighting the flu is a tough task. I liked these tips.
Stay well this flu season. Take a look at this -
So many ways to get sick this holiday season. Stay healthy!
The flu is for real. Do everything you can. Here are some tips:
My sister has the flu right now. Not fun. Save yourself.
Traveling is hard enough without the flu. Stay well and enjoy time off.
It's important to be informed. Here's some information to stop the flu.
I don't want to be sick. Do you want to be sick? I found this useful.
Don't let the flu be the reason you can't enjoy the holiday season.
Take a look at this... some helpful information about preventing the flu.

Appendix B - Filler Tweets

This is the list of hashtags and users that were used to fill the Wilsons' profiles with activity to mask the experiment. Every three minutes, a pythonic coin flip determined whether or not to retweet. If the decision was to proceed with the retweet, one of these hashtags or handles was picked at random with varying probability loosely based on the frequency that the user or hashtag posted. The goal was to provide a continuous supply of tweets that could be retweeted without our intervention. To assist with this, only the last 200 tweets from the chosen hashtag or user that were in english and without the "sensitive" flag set were used for retweeting.

#healthyliving
#healthyrecipes
#healthyeating
#landscapephotography
@MayoClinic
@HikeLoseWeight
@CSPI
@USDANutrition
@JohnsHopkinsSPH
@NPRHealth
@HarvardHealth
@CDCgov
@lifehacker
@NatGeoTravel

Appendix C - Final Pipeline Architecture

The following is a diagram depicting the final serverless architecture of the tweet intake pipeline.