LSTM For Identifying Russian Political Troll-Bots On Twitter

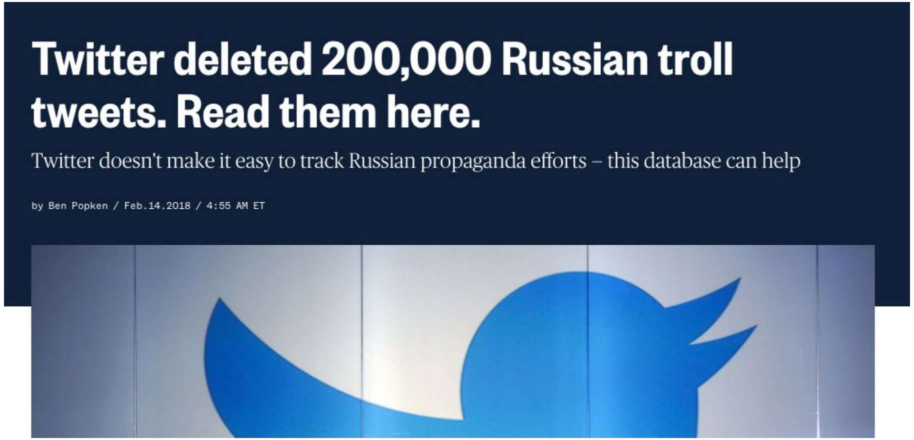
Brandon Cummings / Tako Hisada / Cameron Kennedy

Background and Motivation

- 2017 Google, Facebook, and Twitter testified before congress on Russian interference in US elections.
- Report advertising spend, malicious accounts, and hundreds of disinformation campaigns.

Background and Motivation - Continued

- Twitter confirms 3,814 malicious accounts linked to Internet Research Agency
- Twitter hands over data to House Intelligence Committee
- Feb 14, 2018 NBC releases dataset of 200,000 tweets from 394 accounts


Twitter deleted 200,000 Russian troll tweets. Read them here.

Twitter doesn't make it easy to track Russian propaganda efforts – this database can help

by Ben Popken / Feb.14.2018 / 4:55 AM ET

Research Question

Is it possible to identify an IRA account from the twitter disinformation campaign of the 2016 U.S. Presidential Election?

Datasets

Genuine accounts: cresci-2015 dataset from Bot Repository

Malicious accounts: Dataset published by NBC News article “Twitter Deleted Russian Troll Tweets. So We Published More than 200,000 of Them”

Dataset	Number of Accounts	Number of Tweets
Genuine Accounts	3,475	2,799,999
Malicious Accounts	394	203,482

Datasets - Continued

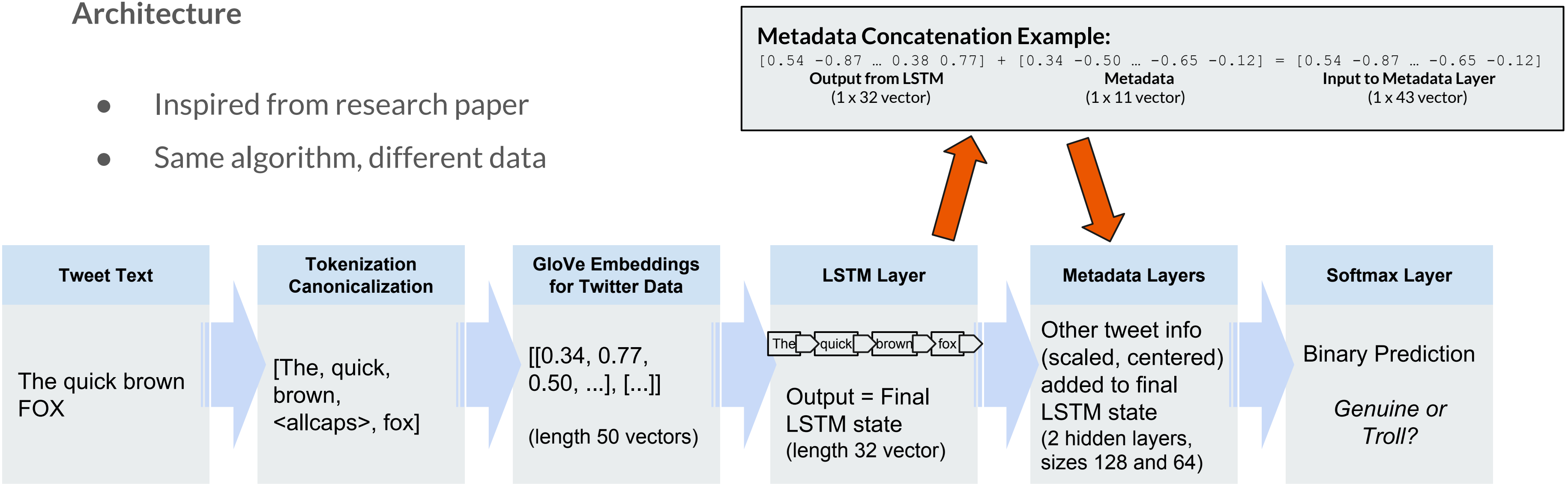
- Text - the actual tweet text
- Metadata:

Tweet Info	Retweet Count - Number of times a tweet has been retweeted Favorite Count - Number of other users that favorited the Tweet Number of Hashtags - Number of hashtags referenced in a Tweet Number of URLs - Number of URLs referenced in a Tweet Number of Mentions - Number other users' handles in the Tweet text Tweet Length - Length of the tweet (count of tokens, project derived)
User Info	Statuses Count - Number of Tweets (including retweets) issued by the user Followers Count - Number of followers this account currently has Friends Count - Number of users this account is following Favourites Count - Number of Tweets this user has liked in the account's lifetime Listed Count - Number of public lists that this user is a member of

Methods

Architecture

- Inspired from research paper
- Same algorithm, different data



Models

Common Across All Models:

- Subset of tweets (340K out of ~3M tweets)
  - 5% of genuine tweets - 140K tweets
  - 100% of troll tweets - 200K tweets
- Ample pre-processing (e.g., handling NAs, hashtags, memory management, unknown tokens)

Model	Description
1. Baseline: Tweet Text Only	Train on tweet text only, no post-LSTM layers
2. Tweet Text + All Metadata	Adds metadata and 2 post-LSTM hidden layers
3. Tweet Text + Select Metadata	Same as #2, with 4 of 11 metadata features removed

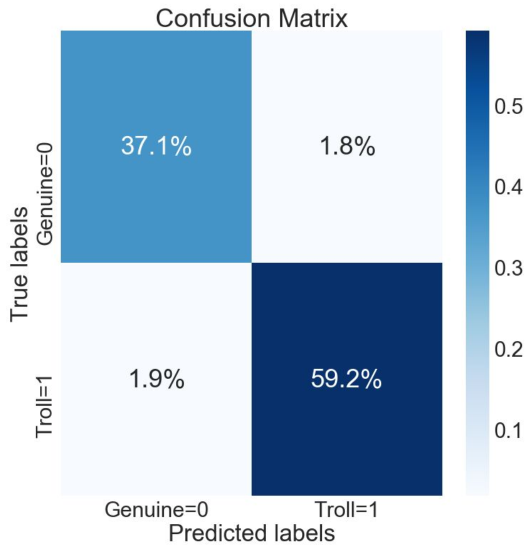
Results / Analysis

Results Summary

Model	Accuracy
1. Baseline: Tweet Text Only	90%
2. Tweet Text + All Metadata	99%
3. Tweet Text + Select Metadata	96%

- Ran several additional models to tweak parameters
  (batch size, epochs, learning rate, # of metadata layers, # of nodes)

Model 3 Results:

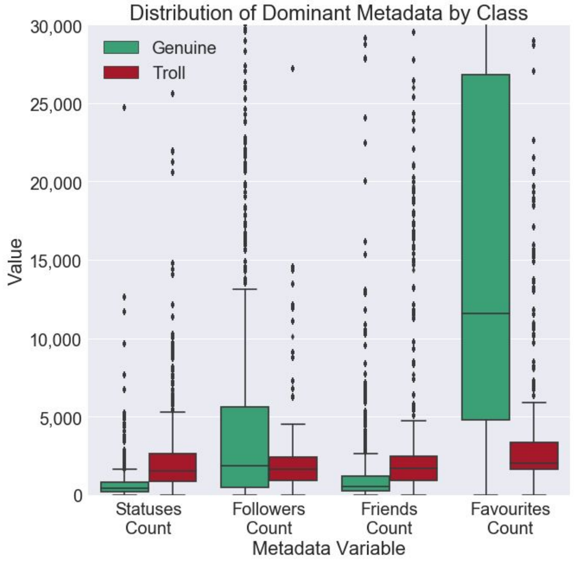


Model 2: 99% Accuracy? That's Suspicious ...

Findings:

- A few common tokens predict most IRA tweets:
  - [<allcaps>, :, rt, <url>, <hashtag>, ...]
  - Biggest: 'rt' (65% difference), 35% still unexplained
- Discovered 4 of 11 metadata fields dominated model
  - Effectively an 'answer key'
  - Eliminates contribution from tweet text
  - Inspired Model 3, balancing text + metadata contributions

Why? 4 Key Metadata Fields



Error Analysis - Summary of Text Findings

Analyzed by True / False Positives / Negatives:

Analysis	Key Findings
Manual inspection of tweets	• URLs common in false positives and negatives • Frequent concatenated words (e.g., 'weirdthingstobuyonline') • Retweets common (tagged, as opposed to merely including 'rt') • Some genuine tweets looked like IRA, and vice versa
Token counts (as %), genuine vs. troll	• Big differences in genuine vs. troll from a few key tokens • Frequent 'rt' despite tweet not flagged as retweet
Unknown tokens (not in GloVe)	• Punctuation errors (fixed custom twitter tokenizer) • Concatenated words
Unique tokens in false positives / negatives	• Frequent politically oriented words in false negatives • Frequent concatenated words (text, hashtags, and usernames)

Next Steps

Remaining Project Tasks for Consideration:

- Train using the full 2.8M tweets (more computing resources required)
  - Might improve metadata domination
  - Requires oversampling the troll tweets (currently considering SMOTE)
- Refine error analysis

Considerations for Future Work

- Incorporate 3M IRA bot dataset published by Oliver Roeder on FiveThirtyEight on July 31, 2018
- Find consistent data and metadata (pull from the same source)
- Find and fix additional tokenizer errors
- Incorporate other tweets (e.g., genuine tweets with similar political content)
- Analyze URL content

Conclusions

Key Takeaways:

- Successfully applied algorithm from different application to new data set
- Models predict IRA Tweets very well, largely aided by metadata and a few key tokens
- Adding metadata to LSTM with text alone is a viable strategy

Q & A

What questions do you have?

Appendix

Error Analysis - Token Prevalence

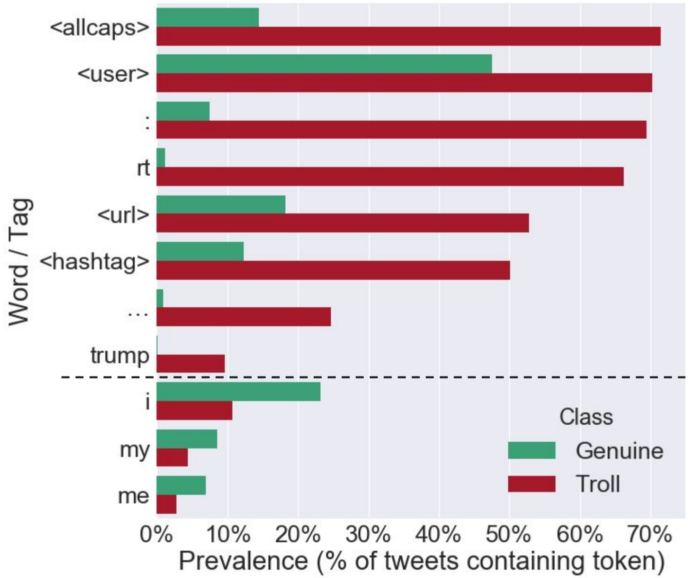
Findings:

- A few common tokens predict most troll tweets:
  - [<allcaps>, :, rt, <url>, <hashtag>, ...]
  - Biggest: ‘rt’ (65% difference), 35% still unexplained

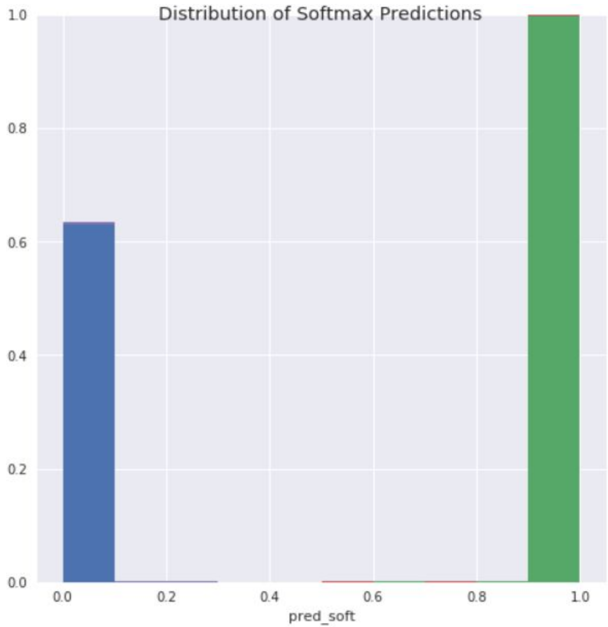
- Top words found more in genuine tweets:
  - [I, my, me]
  - Perhaps Genuine tweets are more personal?

Token Prevalence by Class

(showing tokens with greatest difference between classes)



Error Analysis - SoftMax Prediction Distribution

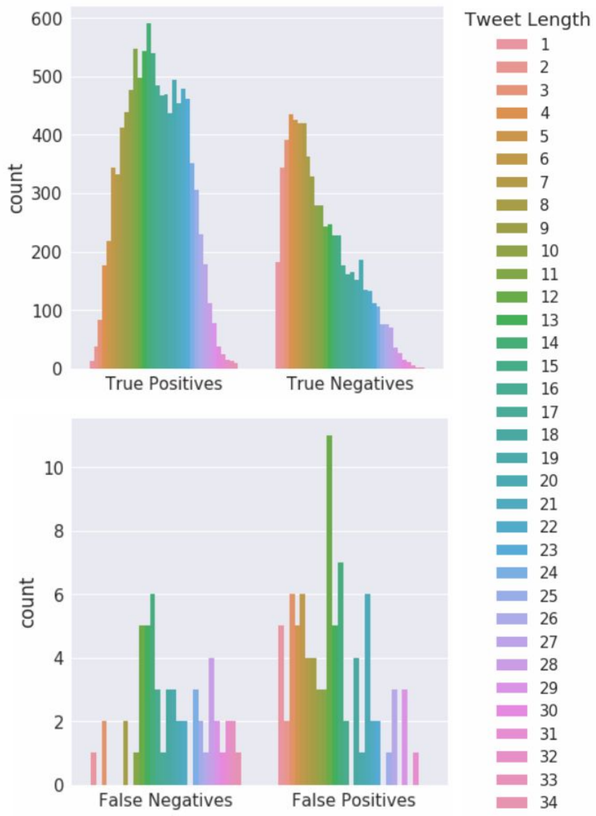


Error Analysis - Tweet Length

Findings:

- True positives appear normally distributed
- True negatives appear left skewed
- False positives and negatives appear roughly normal, but more data would help

Tweet Length by Result



Error Analysis - Tweet Text

False Positives (pred=Troll, target=Genuine)	**False Negatives** (pred=Genuine, target=Troll)
• “@KrizAlin11: Beyonce raking in 50 million more from Pepsi.” To white folk, that's 'selling out', but nbl "gotta make money, shit" • Johnny manziel is just a little punk bitch that needs to grow up and show some class • Steal a moment from the "Destiny". #lifeisforsharing • Are you applying the '10X Rule' to your life and business? Here I discuss using it with blogging - http://t.co/640vzwMe3N • CLICK AND LAUGH your head off! https://t.co/rijhNeRx8d • @DylanNierstedt It gets the people going. • One morning Modi received a threat call that some terror outfit is going to attack on the Republican Day rally .... http://t.co/2KFgiZiB27 • i can't believe we painted all of the elementary schools • @karennngalvann is "aunt" code for you. Don't worry. I get what your saying. We speak in code • this is a special report I wrote on relationship marketing - http://t.co/h81PBFxj - enjoy!	• I see #sick people evryday • #SometimesTwitterMakesMe Want to die because of all the sh*ty updates and changes over the years, at least I can still see pictures now. :| • What do you see when you look into my eyes? #badday • #offline! • The #US Air Force once again delays plans to retire the A-10 Thunderbolt II https://t.co/B4BrWbnh4d • when @realDonaldTrump asks Black People "what do we have to lose?" like it's a game show...respond with " #OurLives" https://t.co/8JiEZK9GvW • Poll: @HillaryClinton hits new high in unpopularity. https://t.co/gApVxNIvtK • RHONYs Kristen Taekman husband Josh is an Ashley Madison user • .@HillaryClinton Yeah, right https://t.co/QKgSjtARhU • @realDonaldTrump bruh you can't even handle AC mobsters and you gonna handle Isis? https://t.co/22UjgqBu1f

Error Analysis - Non-Common Words

Top Words Not Common to All Lists			
False Positives (pred=1, targ=0)	**False Negatives** (pred=0, targ=1)	**True Positives** (pred=1, targ=1)	**True Negatives** (pred=0, targ=0)
one	what	…	just
just	think	trump	but
❤	star	’	was
thanks	👌	they	what
into	black	by	no
<blank>	want	what	<smile>
sleep	week	obama	<elong>
<smile>	going	hillary	one
know	say	who	good
going	they	as	out

Is This Model Just a Word Finder?

Were political words a big factor? Somewhat, but most performance came from non-political words.

- Political words were much more common in troll tweets (29% troll vs. 1% genuine)
- But that leaves 71% of troll tweets without political words
- Political words: ['trump', 'donald', 'hillary', 'clinton', 'bernie', 'sanders', 'obama', 'bush', 'election', 'vote', '2018', 'polit', 'islam', 'muslim', 'washington', 'president', 'country']

Error Analysis - Results by Epoch and Timing

(Model 3)

```
Number of batches: 450
Epoch - 1: Error = 2.63%.  Time to train 1 epoch(s): 48 seconds
Epoch - 2: Error = 1.87%.  Time to train 2 epoch(s): 100 seconds
Epoch - 3: Error = 1.51%.  Time to train 3 epoch(s): 152 seconds
Epoch - 4: Error = 1.31%.  Time to train 4 epoch(s): 203 seconds
Epoch - 5: Error = 1.37%.  Time to train 5 epoch(s): 254 seconds
Epoch - 6: Error = 1.46%.  Time to train 6 epoch(s): 305 seconds
Epoch - 7: Error = 1.40%.  Time to train 7 epoch(s): 355 seconds
Epoch - 8: Error = 1.34%.  Time to train 8 epoch(s): 407 seconds
Epoch - 9: Error = 1.12%.  Time to train 9 epoch(s): 455 seconds
Epoch - 10: Error = 1.09%.  Time to train 10 epoch(s): 505 seconds
Epoch - 11: Error = 1.06%.  Time to train 11 epoch(s): 556 seconds
Epoch - 12: Error = 1.14%.  Time to train 12 epoch(s): 607 seconds
Epoch - 13: Error = 1.09%.  Time to train 13 epoch(s): 658 seconds
Epoch - 14: Error = 0.95%.  Time to train 14 epoch(s): 709 seconds
Epoch - 15: Error = 0.94%.  Time to train 15 epoch(s): 759 seconds
Epoch - 16: Error = 0.91%.  Time to train 16 epoch(s): 808 seconds
Epoch - 17: Error = 0.82%.  Time to train 17 epoch(s): 857 seconds
Epoch - 18: Error = 0.86%.  Time to train 18 epoch(s): 905 seconds
Epoch - 19: Error = 0.84%.  Time to train 19 epoch(s): 954 seconds
Epoch - 20: Error = 0.84%.  Time to train 20 epoch(s): 1003 seconds
Fetch numerous tensors for post hoc analysis ... done!
Time to run cell: 1004 seconds
```

Error Analysis - Tuning Log

Results Log

ID	Date	Time (MDT)	Genuine Tweets	Russian Tweets	GloVe Size	LSTM Cell Size	Metadata Used	Epochs	Batch Size	Learning Rate	Adam. Epsilon	Train Size	Dev Size	Test Size	Metadata Layers: [Sizes]	Variable Length	Over-sampling	Fixed Punctuation Tokenization	Accuracy Tested On	Accuracy (final epoch)	Training Time (s)	Training Time (m)	Time / Epoch (s)	Comments
6	7/25/2018	5:30 PM	139,376	203,482	50	77	Tweet counts & user counts	50	500	Default (0.0010)	Default (1e-08)	250,000	25,000	26,000	None	No	No	No	Test	96.4%	4,156	69	83	96% after Epoch 11; plateaued at epoch 24
7	7/26/2018	8:10 PM	139,376	203,482	50	34	Tweet counts & user counts	10	500	Default (0.0010)	Default (1e-08)	250,000	25,000	30,000	None	Yes	No	No	Test	96.0%	514	9	51	First time with variable length. Interesting that it takes a bit longer than with fixed length.
8	7/26/2018	8:25 AM	139,376	203,482	50	34	Tweet counts & user counts	20	500	Default (0.0010)	Default (1e-08)	250,000	25,000	30,000	None	Yes	No	No	Test	96.4%	1,066	18	53	No better than without variable length. Did I implement variable length correctly? Also, ran 2nd time with state[0], same result.
9	7/26/2018	9:55 PM	139,376	203,482	50	34	Tweet counts & user counts	35	500	Default (0.0010)	Default (1e-08)	250,000	25,000	30,000	None	Yes	No	No	Test	96.8%	1,706	28	49	Maybe dynamic length helped? Ran with state[0].
10	7/26/2018	10:25 PM	139,376	203,482	50	34	Tweet counts & user counts	50	500	Default (0.0010)	Default (1e-08)	300,000	25,000	17,837	None	Yes	No	No	Test	96.6%	2,911	49	58	
11	7/28/2018	9:50 PM	139,376	203,482	50	34	Tweet counts & user counts	10	500	Default (0.0010)	Default (1e-08)	300,000	25,000	17,837	1: [57]	Yes	No	No	Test	96.5%	605	10	61	Adding metadata hidden layer dramatically sped up performance (arrived at similar results in ~half the time).
12	7/29/2018	6:00 AM	139,376	203,482	50	34	Tweet counts & user counts	20	500	Default (0.0010)	Default (1e-08)	300,000	25,000	17,837	1: [64]	Yes	No	No	Test	96.5%	1,290	22	65	Odd we didn't see improvement
13	7/29/2018	6:40 AM	139,376	203,482	50	32	Tweet counts & user counts	20	500	Default (0.0010)	Default (1e-08)	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Dev	96.1%	1,163	19	58	Divergence! Epoch 1 error was 5.41%, 2 was 6.62%, 3 was 6.85%. 4 dropped to 5.27%, 5 up to 5.72%
14	7/29/2018	7:10 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0005	Default (1e-08)	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Test	94.5%	599	10	60	
15	7/29/2018	7:40 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0100	Default (1e-08)	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Test	96.3%	594	10	59	Accurady still bouncing around between epochs. Learning rate doesn't seem to affect this. Metadata layers might be the culprit.
16	7/29/2018	8:05 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0010	1E-04	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Test	95.9%	598	10	60	Highly erratic accuracies between epochs.
17	7/29/2018	8:25 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0010	1E-10	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Test	95.5%	591	10	59	Let's see if decreasing epsilon helps the bouncing ... nope, still bouncing. Time to adjust metadata layers.
18	7/29/2018	8:38 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0010	1E-10	300,000	25,000	17,837	2: [64, 128]	Yes	No	No	Test	95.5%	584	10	58	Still bouncing. Might be just because there are 2 layers
19	7/29/2018	9:00 AM	139,376	203,482	50	32	Tweet counts & user counts	10	500	0.0010	1E-10	300,000	25,000	17,837	1: [128]	Yes	No	No	Test	96.7%	589	10	59	Ah ha! So that 2nd metadata layer seems to clearly be the cause of the bouncing. Let's try this with 20 epochs.
20	7/29/2018	9:30 AM	139,376	203,482	50	32	Tweet counts & user counts	20	500	0.0010	1E-10	300,000	25,000	17,837	1: [128]	Yes	No	No	Test	96.3%	1,151	19	58	
21	7/29/2018	1:00 PM	139,376	203,482	50	32	Tweet counts & user counts	50	2,000	0.0010	1E-08	300,000	25,000	17,837	2: [128, 64]	Yes	No	No	Test	96.5%	2,356	39	47	Trying larger batch size with two metadata hidden layers. Still some bouncing. Not worth the training time.
22	7/29/2018	2:00 PM	139,376	203,482	50	32	Tweet counts & user counts	2	500	0.0010	1E-08	300,000	25,000	17,837	1: [64]	Yes	No	No	Test	95.8%	116	2	58	Look how close this is with only 2 epochs. Could be part luck, but though it worth noting.
23	8/1/2018	Morning	103,178	163,810	50	32	Tweet counts, user counts, excl. RT	10	500	0.0010	1E-08	225,000	25,000	16,988	1: [64]	Yes	No	No	Test	97.6%	454	8	45	Removed retweets from data
24	8/1/2018	9:30 PM	103,178	163,810	50	32	Tweet counts, user counts, excl. RT	10	500	0.0010	1E-08	225,000	25,000	16,988	1: [64]	Yes	No	No	Test	97.4%	459	8	46	Added ReLU! Notably, Dev set accuracy was 98.6%. Not sure this 97.4% result is representative; think it might actually be better.
25	8/1/2018	9:40 PM	103,178	163,810	50	32	Tweet counts, user counts, excl. RT	10	500	0.0010	1E-08	225,000	25,000	16,988	2: [128, 64]	Yes	No	No	Test	98.9%		0	0	2nd layer helped, and we've now solved the problem we had above. Some quick searching suggest it was the 'vanishing gradient problem,' which the ReLU solved.
26	8/1/2018	10:20 PM	103,178	163,810	50	32	Tweet counts, user counts, excl. RT	20	500	0.0010	1E-08	225,000	25,000	16,988	2: [128, 64]	Yes	No	No	Test	99.3%	936	16	47	Same as above but with 20 epochs.
27	8/2/2018	12:00 PM	103,178	163,810	50	32	Tweet counts, user counts, excl. RT	20	500	0.0010	1E-08	225,000	25,000	16,988	2: [128, 64]	Yes	No	Yes	Test			0	0	Fixed punctuation tokenization resulting in many fewer unknown (to GloVe) words.

Error Analysis - Summary of Findings

Word Errors	Non-Word Errors
● Removing retweets ● Unknown words: ○ Custom twitter tokenizer not splitting punctuation (e.g., “<user>:”) ○ Multiple words / tags without spaces (e.g., hillaryclinton, <number>th) ○ Missed several emojis ● Custom twitter tokenizer splitting contractions ● Unique words in false positives / negatives (not common across all lists) ● False positives often contain URLs ● Highly plausible of true class / uncharacteristic of predicted class	● Metadata (main premise of paper) ○ Tweet text ○ Tweet length ○ Mostly 0’s ● Sequence Length ● ReLU Activations fixed ● Full data set