COVID-19 Retweet Prediction

Aaron Khoo, Calvin Yusnoveri, Amarjyot Kaur Narula, Joseph Chng

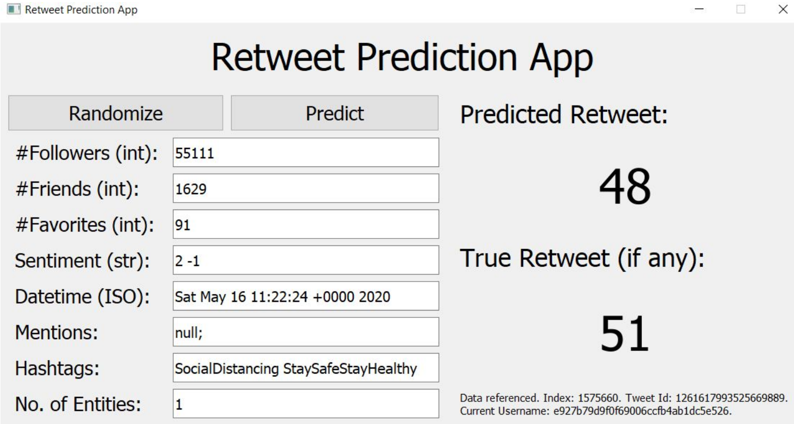
Content

1. Task Description
2. Dataset Description
3. Preprocessing (final data structure)
    a. Hashtags & Mentions
    b. Timestamp
    c. Sentiment
    d. No. of Entities
    e. User Context
4. Model Architecture
5. Results
    a. Hyperparameters
    b. Result
    c. Accuracy
6. Discussions

Task Description

- Measure the “impact” of COVID-19 through Twitter activity

- Predict the “ballpark” Retweet count of a Tweet based on several input parameters as found in TweetsCOV-19 dataset.



Dataset Description

The dataset that is used for this project is obtained from the COVID-19 Retweet Prediction Challenge, here: https://data.gesis.org/tweetscov19/#dataset. The fields are:

1. Tweet Id: Long. Unique ID for a specific tweet
2. Username: String. Username of the user that published the tweet which is encrypted for privacy.
3. Timestamp: Format ("EEE MMM dd HH:mm:ss Z yyyy"). Specific time and date of the tweet
4. #Followers: Integer. Number of followers of the Twitter user who posted the tweet.
5. #Friends: Integer. Number of friends that the Twitter user who posted the tweet.
6. #Retweets: Integer. Number of retweets that the tweet has obtained and is the label for this project.
7. #Favorites: Integer. Number of favorites for the tweet

Dataset Description

8. Entities: String. The entities of the tweet is obtained by aggregating the original text. Every annotated entity will then have its produced score from FEL library. Each entity is separated by char ":" to store the entity in this form "original_text:annotated_entity:score;". Each entity is separated from another entity by char ";".Any tweet that has no corresponding entities will be stored as "null;".
9. Sentiment: String. SentiStrength produces a score for positive (1 to 5) and negative (-1 to -5) sentiment. The two sentiments are splitted by whitespace char " ". Positive sentiment was stored first and followed by negative sentiment (i.e. "2 -1").
10. Mentions: String. Contains mentions and concatenate them with whitespace char " ". If there is no mention, it is stored as "null;".
11. Hashtags: String. Contains hashtags and concatenate the hashtags with whitespace char " ". If there is no hashtag, it is stored as "null;".
12. URLs: String: Contains URLs and concatenate the URLs using ":-: ". If there is no URL, it is stored as "null;"

Preprocessed Data

Due to constraints such as GPU and Memory, we train over a smaller subset of the data randomly from the source. The final preprocessed data has the following input fields < dim: (77,) >:

The target is: #Retweets: Integer. Log transformed: log_10(x + 1)

1. #Followers: Float. Log transformed: log_10(x + 1)
2. #Friends: Float. Log transformed: log_10(x + 1)
3. #Favorites: Float. Log transformed: log_10(x + 1)
4. Positive (Sentiment): Float. Scaled.
5. Negative (Sentiment): Float. Scaled.
6. Sentiment Disparity: Float. Scaled.
7. No. of Entities: Float. Log transformed: log_10(x + 1)

Preprocessed Data

8. Day of Week: Float. One-Hot Vector. (7,) Vector.
9. Time Int: Float. Log transformed: log_10(x + 1)
10. Hashtags Embedding: (25,) Vector.
11. Mentions Embedding: (25,) Vector.
12. #Followers Min, Max, Mean: Float. Log transformed: log_10(x + 1)
13. #Friends Min, Max, Mean: Float. Log transformed: log_10(x + 1)
14. #Retweets Min, Max, Mean: Float. Log transformed: log_10(x + 1)
15. #Favorites Min, Max, Mean: Float. Log transformed: log_10(x + 1)

In total, the input dimension (first layer) is 8 + 7 + 25 + 25 + 12 = (77,).

Preprocessing: Hashtag & Mention

- In order to create tractable input for the model, embeddings are created for both the Hashtags and Mentions of size (25,)
- Trained over 5 epochs and only considers symbols that appear more than 200 times
- “Null” cells are 0 vector
- Out of vocab symbols are considered “Null” as well
- Multiple symbols’ vectors are summed

```
print(hashtags_vocab[:5]) # example of hashtags key
print(mentions_vocab[:5]) # example of mentions key
```

```
['COVID19', 'coronavirus', 'Covid_19', 'covid19', 'May']
['realDonaldTrump', 'PMOIndia', 'narendramodi', 'jaketapper', 'YouTube']
```

```
hashtags_example = 'COVID19'
mentions_example = 'realDonaldTrump'

print(f"{hashtags_example} -> {hashtags_embeddings.wv[hashtags_example]}")
print(f"{mentions_example} -> {mentions_embeddings.wv[mentions_example]}")
```

```
COVID19 -> [ 0.10622272  0.26996937 -0.46450084  0.10561462 -0.5595082   0.26207525
  0.28835535  0.80339587  0.30626374 -0.13036335  0.8120623  -0.46314418
  0.20126966 -0.9723947  -1.0051426  -0.04809839  0.4593365   0.09532893
 -0.21894015 -0.23557915  0.42107382  0.4622469   0.53460604 -0.589559
  0.6296402 ]
realDonaldTrump -> [ 0.5991303  -0.10410535  0.23690729 -0.23115875 -0.961905   -0.11418784
  0.12405131  0.4196795  -1.493182   -0.20270342  1.2276924  -1.3593616
 -0.19556278  0.27365074  0.32451993  1.9415929  -0.20647514 -0.17526582
 -0.69910485 -1.6436449   1.3161302  -0.17269903 -0.5424232   1.0386076
  1.062889  ]
```

Preprocessing: User Context

- Provide contextual information about the type of user for each post
- Look Up table, input to NN
- **Minimum, maximum, mean values** of No. of followers, No. of friends, No. of Retweets & No. of favourites.
- Applied logarithmic transformation

Preprocessing: Timestamp

1. Day of the Week

- Integer from 0 to 6, 0=Sunday, 1=Monday,...
- Converted to one-hot vector of length=7
- Unpacked into 7 new columns for each day

2. Time Integer

- Integer number of seconds from 1st Jan 2019 to each datetime.
- Applied logarithmic transformation to remove skewness

Preprocessing: Sentiment

- Positive: 1 to 5, Negative: -1 to -5
- Extracted Positive and Negative scores as separate features
- Applied Scaled Transformation by the mean value.

Preprocessing: No. of Entities

- Encapsulated in the dataset from the original tweet text
- Text goes through a Fast Entity Linker query and find annotated text that can be found and set them as an entity
- For every entity, it also has its corresponding log-likelihood confidence score which is used as a global threshold for linking
- Obtain the number of entities that is found on each tweet and undergo logarithmic transformation of the form $\log(x+1)$ before they're passed into the model

Model Architecture

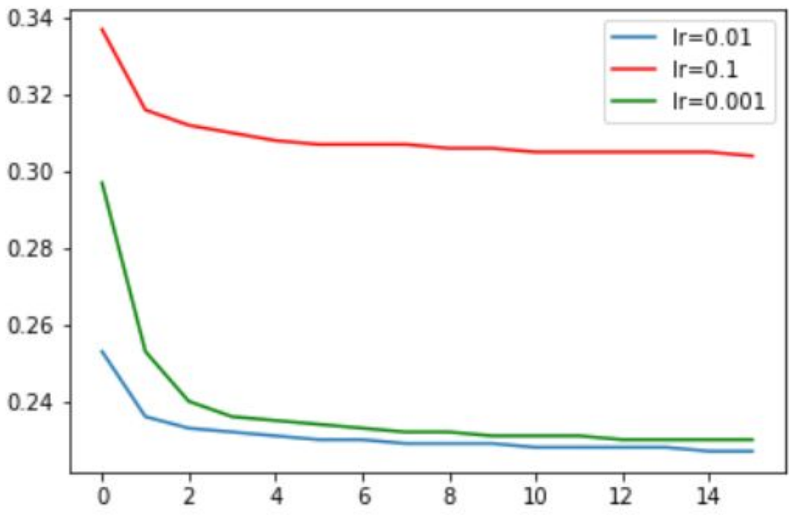
- (Final) Simple 2 Layer Regression model
  - More layers were tested without significant improvement
  - Avoids over-parameterized model which can overfit
- XGBoost tested, but weaker performance
- 2-step model tested:
  - Classifier > Regression, but Classifier is difficult to train

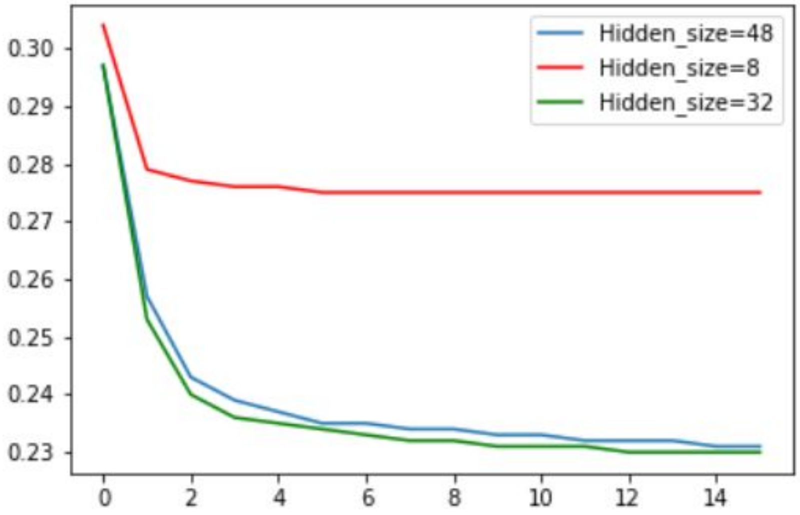
```
# Hyperparameters
input_size = 77
hidden_size = 32
output_size = 1
learningRate = 0.01

model = LinReg2(input_size, hidden_size, output_size)
model
```

```
LinReg2(
  (fc1): Linear(in_features=77, out_features=32, bias=True)
  (relu_h1): ReLU()
  (fc2): Linear(in_features=32, out_features=1, bias=True)
)
```

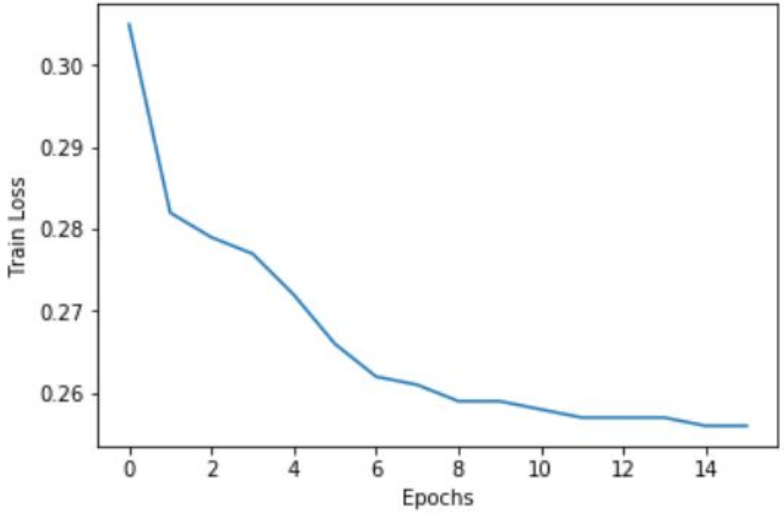
Results: Hyperparameters





Results: Training Loss

- Given that our training model has its training loss plateauing at around 15 epochs.
- We have also picked 15 as the number of epochs that our model would be trained.
- Accordingly, the following hyperparameters are chosen:
    - Learning rate = 0.01
    - Hidden size = 32



Results: Accuracy

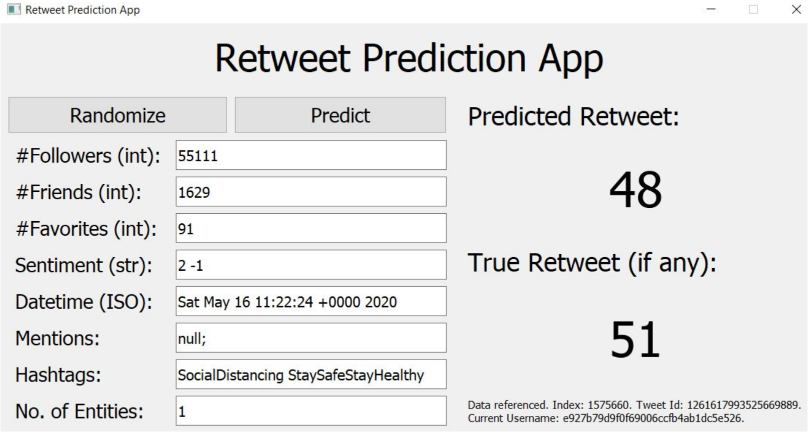
- To measure the accuracy of the model, we take the log10 of the absolute difference between the prediction and true retweet value + 1.
    - log10(|y - t| + 1)
- The worst deviation is around 4.72 order of magnitude away (difference of 50,000 retweets).
    - Extremely rare, as a deviation of 4.0+ occurs less than 0.01% of the time in the test set.
- The model does pretty well in most cases (~97% of cases),
    - 79.4% of test data prediction lands within 0.0 - 0.99 order of magnitude of the actual tweet (difference of 10 retweet).
    - 17.6% of next portion of test data prediction falls within 1.00 - 1.99 order of magnitude (difference of 100 retweet).

```
Min Order of Magnitude Deviation: 0.0
Mean Order of Magnitude Deviation: 0.4959347980368691
Median Order of Magnitude Deviation: 0.4959347980368691
Max Order of Magnitude Deviation: 4.7193975864238835
```

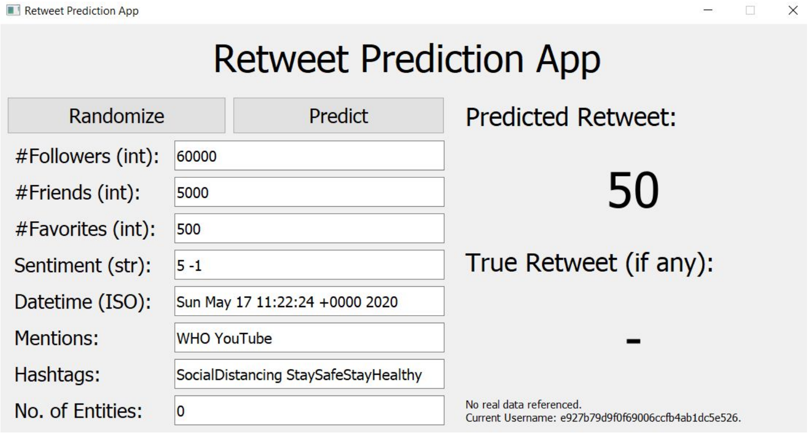
```
{0.0: 79.44831353270517,
 1.0: 17.647862094448595,
 2.0: 2.72469495786844,
 3.0: 0.1694901639475946,
 4.0: 0.009639251030194954}
```

GUI

Using Dataset



Using Custom Data



Discussions

- Comparison with state-of-the-art
  - Ensemble model with classifier before regression
    - 20/80 split, k-means clustering, and 0-1 labelling
  - User context via attention mechanism
    - Lookup table based on username
- Improvements
  - Attention layer instead of bulky lookup table
  - Selection of different loss function
    - Absolute difference vs MSE
  - Better feature engineering, such as embedding on entities
    - Winning paper uses 500 - higher cardinality and more features engineered