Project 5: Popularity Prediction on Twitter

Vignesh Muralidharan (904729596)
Chidambaram Muthappan (704774938)
Jeya Vikranth Jeyakumar (404749568)

1 Solution 1

Our task is to explore the training tweet data and calculate some important statistics for each hashtag, such as the average number of tweets per hour, average number of followers of users posting the tweets and average number of retweets.

The hashtags that were provided in the training data were #gohawks, #gopatriots, #nfl, #patriots, #sb49, #superbowl. Here are the following statistics that we obtained for each of the hashtags:

#gohawks

Average tweets/hour: **193.545**

Average retweets/hour: **2.015**

Average num. of followers: **2393.582**

#gopatriots

Average tweets/hour: **38.385**

Average retweets/hour: **1.400**

Avg num. of followers: **1602.01**

#nfl

Average tweets/hour: **279.551**

Average retweets/hour: **1.539**

Avg num. of followers: **4763.326**

#patriots

Average tweets/hour: **499.421**

Average retweets/hour: **1.783**

Avg num. of followers: **3641.688**

#sb49

Average tweets/hour: **1419.888**

Average retweets/hour: **2.511**

Avg num. of followers: **10230.045**

#superbowl

Average tweets/hour: **1401.246**

Average retweets/hour: **2.388**

Avg num. of followers: **9958.116**

In a tabular format, the following values can be represented as follows:

Table 1: The data information for each hashtag

	gohawks	gopatriots	nfl	patriots	sb49	superbowl
Average number of tweets per hour	193.545	38.385	279.551	499.421	1419.888	1401.246
Average number of follower per users	2393.582	1602.01	4763.326	3641.688	10230.045	9958.116
Average number of retweets per hour	2.015	1.400	1.539	1.783	2.511	2.388

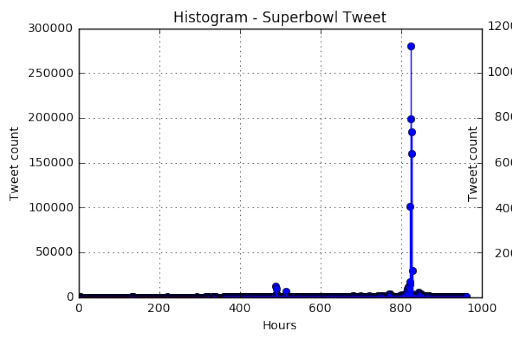
1.1 Procedure

We parsed each line, i.e. an individual tweet, contained within a hashtag file as a JSON object called as individual tweet data. By using the Twitter API documentation as a reference, we then extracted the information related to the attributes of our interest from the tweet data as follows:

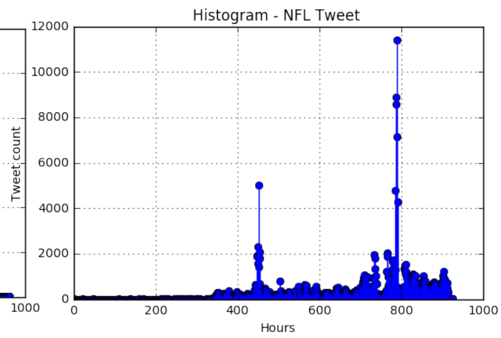
- followers = *individual_tweet_data*["tweet"]["user"]["followers_count"]
- date = *individual_tweet_data*["firstpost_date"]
- retweets = i*ndividual_tweet_data*["metrics"]["citations"]["total"]

Following this step, we computed the appropriate desired counts from the attributes given above, we calculated the statistics mentioned in the question using the following formulae:

$$\text{Average no. of tweets per hour} = \frac{\text{Total number of tweets}}{\text{Total number of hours elapsed}} \tag{1}$$



(a) Histogram for Superbowl over time (one hour bins)



(b) Histogram for NFL over time (one hour bins)

$$\text{Average no. of followers of users} = \frac{\text{Total number of followers}}{\text{Total number of tweets}} \tag{2}$$

$$\text{Average no. of retweets} = \frac{\text{Total number of retweets}}{\text{Total number of tweets}} \tag{3}$$

1.2 Observations

- It is clearly evident that **#sb49** and **#superbowl** have the most number of tweets per hour, the most number of retweets per tweet and the most average number of followers. This might be because hashtags like **#sb49** and **#superbowl** are "common" hash- tags and all the people watching the Superbowl are likely to be tweet them and hence they have a high chance of becoming burst hashtags. In other words, they also

- In addition to it, hashtags like **#gopatriots** and **#gohawks** are team specific and are not likely to get retweeted by the entire population watching the game but rather only specific sections and supporters. Therefore, in those cases, the average tweets per hour and average number of followers is **low** compared to the other popular hashtags.

Histograms representing number of tweets in one hour over time for #nfl and #superbowl are shown in Figures 1 and 2.

1.3 Observations from Histograms

From figures 1 and 2, histograms clearly indicate that there are peaks around one of the 800th hours for both the **#nfl** and **#superbowl**. This is the burst time of the tweets and we believe that this might correspond to the hours in which the **superbowl match might have actually taken place** and hence the very high number of tweets. Specifically, the NFL tweet histogram has its peak little less than its 800th hour, whereas Superbowl tweet histogram has its peak little above 800th hour.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.734
Model:                            OLS   Adj. R-squared:                  0.733
Method:                 Least Squares   F-statistic:                     1962.
Date:                Mon, 14 Mar 2016   Prob (F-statistic):               0.00
Time:                        00:28:35   Log-Likelihood:                -35003.
No. Observations:                3566   AIC:                         7.002e+04
Df Residuals:                    3560   BIC:                         7.006e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        123.4339    149.431      0.826      0.409      -169.546   416.414
x1             0.4857      0.056      8.610      0.000         0.375     0.596
x2             0.4632      0.038     12.324      0.000         0.390     0.537
x3            -0.0001    5.8e-06    -19.393      0.000        -0.000    -0.000
x4             0.0003   3.51e-05      8.961      0.000         0.000     0.000
x5           -11.2608     10.991     -1.025      0.306       -32.810    10.288
==============================================================================
Omnibus:                     5987.946   Durbin-Watson:                   2.007
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         52637610.715
Skew:                          10.236   Prob(JB):                         0.00
Kurtosis:                     597.848   Cond. No.                     1.23e+08
==============================================================================
```

Figure 2: OLS Regression Statistics for features from Section 2

2 Solution 2

A linear regression for predicting the number of tweets in the next hour was built with the features of the tweets from the previous hour as the features. The features we considered for our algorithm were the following: number of tweets, total number of retweets, sum of the number of followers of the users posting the hashtag, maximum number of followers of the users posting the hashtag, and time of the day.

The reference we chose was from January 1st, 2015 1 a.m. From this timestep, every other time slot was calculated by converting it to hours. Hence there are totally five columns in our feature matrix (six including the offset column). And our predictant vector will basically be the number of tweets for the next time step i.e (current hour + 3600 seconds).

We tried two different approaches here. In approach one, we used data from all the hash- tags to run a single OLS regression. This gave us very good training model, though our intuition said it wouldn't. In the second approach, we try a different linear model for each hash-tag separately.

2.1 Approach 1

We tried fitting one linear model for all the hashtags and got very good performance. We used the OLSregression toolkit in Python and observed the following results when we t the feature vector to the predictant (described above).

In Figure (3)

- x1 is the number of tweets

- x2 is the total number of retweets
- x3 is the sum of the followers of the users posting the hashtag
- x4 is the maximum number of followers of the user posting the hashtag
- x5 is the time of day.

p-value explanation

In statistics, the p-value is a function of the observed sample results (a test statis- tic) relative to a statistical model, which measures how extreme the observation is. In our case, a very low p-value for a feature (less than 0.05) means that there is enough evidence against the null hypothesis to reject it and we can thus claim that this feature plays an important role in the performance of the model.

Given a null hypothesis for the probability distribution of the data, the p-value can be said to be the probability that the outcome would be atleast as extreme or probably even more extreme than the outcome that was observed. As the p-value goes on decreasing, the evidence against this null hypothesis keeps increasing. This means that we would want to consider the features with the lowest p-values to be most significant.

t-value explanation

The t-statistic, is a ratio of the departure of an estimated parameter from its notional value and its standard error. T-test is basically used for assessing how statistically significant a particular explanatory variable is. It is important to find such significant explanatory variables in order to fit the regression model most efficiently. In other words, the greater the magnitude of T (it can be either positive or negative), the greater the evidence against the null hypothesis that there is no significant difference. The closer T is to 0, the more likely there isn't a significant difference.

In this question, features with a high magnitude of t-statistic will be considered significant and features with t-statistics magnitude closer to zero are not likely to be significant.

Looking at the p-value statistics from figure 3:

It is evident that **number of tweets, total number of retweets, sum of the followers of the users posting the hashtag, maximum number of followers of the user posting the hashtag** are the most significant features due to their very low p-values. One important thing to note here is that all of them are "user" dependent features.

From the t-statistics:

sum of the followers of the users posting the hashtag appears to have the high- est t-statistics magnitude followed by **total number of retweets,**

maximum number of followers of the user posting the hashtag and **number of tweets**.

From the p-value and t-value statistics, we can confidently say that these features contributed most to the model.

Training accuracy of the model

In order to obtain the training accuracy of the linear regression model or observe how closely our model t the data, we can look at the R-squared statistic. R-squared is a statistical measure of how close the data are to the tted regression line. Hence, a higher R-squared value means that the model is able to explain all the variance obtained in the training data. It is a measure of how well your model has fit the data. A value of '1' indicates a perfect fit. Although it should not be '1'. This indicates that your model is over fitting the data.

For our regression model, we got a R-squared value of **0.734** (Figure 3) which we feel is a pretty- good. Also, we should note that the training accuracy is not too high, because in that case, the model might be overfitting the data and need not necessarily generalize well for prediction.

2.2 Approach 2

Intuitively, it could be wise to guess that different types of tweets (segregated by their hashtags) should follow similar "behaviour" like the ones in Approach 1. Here, we fit different linear models for each hashtag data separately. We used the OLSregression toolkit in Python to do OLS regression and we tabulate the result as follows. We use five features for our linear model. In all of our regressions models in this section,

- x1 is the number of tweets
- x2 is the total number of retweets
- x3 is the sum of the followers of the users posting the hashtag
- x4 is the maximum number of followers of the user posting the hashtag
- x5 is the time of day

Linear model for #gohawks

Looking at the t-value and p-value (we consider features with higher t-value magnitude and lower p-value to be signicant), we can say **number of tweets, total number of retweets** and **sum of followers of the users posting the hashtag** to be significant features. For our regression model, R-squared value is **0.467**, which isn't a very good value. Reason could possibly be that we don't have large number of data-points to train or that a linear model can't explain the data well.

Linear model for #gopatriots

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.467
Model:                            OLS   Adj. R-squared:                  0.464
Method:                 Least Squares   F-statistic:                     154.5
Date:                Tue, 21 Mar 2017   Prob (F-statistic):          7.17e-118
Time:                        01:27:26   Log-Likelihood:                -7201.0
No. Observations:                 889   AIC:                         1.441e+04
Df Residuals:                     883   BIC:                         1.444e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const         70.1859     52.086      1.347      0.178       -32.041   172.413
x1             1.2420      3.915      0.317      0.751        -6.441     8.925
x2             0.5885      0.118      5.000      0.000         0.357     0.819
x3             0.0122      0.039      0.313      0.755        -0.064     0.089
x4          7.452e-05   6.64e-05      1.122      0.262     -5.58e-05     0.000
x5            -0.0003      0.000     -2.084      0.037        -0.001 -1.53e-05
==============================================================================
Omnibus:                     1582.824   Durbin-Watson:                   2.309
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          3273204.595
Skew:                          11.385   Prob(JB):                         0.00
Kurtosis:                     299.390   Cond. No.                     4.59e+06
==============================================================================
```

(a) OLS Regression Statistics for #GoHawks

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.775
Model:                            OLS   Adj. R-squared:                  0.773
Method:                 Least Squares   F-statistic:                     467.1
Date:                Mon, 20 Mar 2017   Prob (F-statistic):          7.73e-217
Time:                        20:13:11   Log-Likelihood:                -4316.6
No. Observations:                 684   AIC:                             8645.
Df Residuals:                     678   BIC:                             8672.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const          8.7617      9.992      0.877      0.381       -10.857    28.381
x1             0.3028      0.742      0.408      0.683        -1.154     1.760
x2             0.4770      0.095      5.010      0.000         0.290     0.664
x3           -20.2031      1.055    -19.150      0.000       -22.275   -18.132
x4             0.0007    8.4e-05      8.319      0.000         0.001     0.001
x5            -0.0006     0.0001     -6.212      0.000        -0.001    -0.000
==============================================================================
Omnibus:                      821.199   Durbin-Watson:                   2.313
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           532840.872
Skew:                           5.090   Prob(JB):                         0.00
Kurtosis:                     139.354   Cond. No.                     8.47e+05
==============================================================================
```

(b) OLS Regression Statistics for #GoPatriots

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.568
Model:                            OLS   Adj. R-squared:                  0.564
Method:                 Least Squares   F-statistic:                     160.4
Date:                Tue, 15 Mar 2016   Prob (F-statistic):          9.73e-109
Time:                        16:30:04   Log-Likelihood:                -4788.5
No. Observations:                 617   AIC:                             9589.
Df Residuals:                     611   BIC:                             9616.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        125.0583     46.708      2.677      0.008        33.331   216.786
x1             0.8101      0.164      4.927      0.000         0.487     1.133
x2            -0.2208      0.080     -2.766      0.006        -0.378    -0.064
x3          7.524e-05   2.57e-05      2.933      0.003      2.49e-05     0.000
x4         -7.104e-05   3.57e-05     -1.988      0.047        -0.000 -8.55e-07
x5            -0.6792      3.375     -0.201      0.841        -7.307     5.948
==============================================================================
Omnibus:                      706.859   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           419308.584
Skew:                           4.603   Prob(JB):                         0.00
Kurtosis:                     130.379   Cond. No.                     9.26e+06
==============================================================================
```

(a) OLS Regression Statistics for #nfl

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.689
Model:                            OLS   Adj. R-squared:                  0.687
Method:                 Least Squares   F-statistic:                     394.7
Date:                Tue, 21 Mar 2017   Prob (F-statistic):          3.60e-223
Time:                        21:24:28   Log-Likelihood:                -8100.2
No. Observations:                 898   AIC:                         1.621e+04
Df Residuals:                     892   BIC:                         1.624e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        132.5815    130.876      1.013      0.311      -124.278   389.441
x1            -2.6816      9.720     -0.276      0.783       -21.759    16.395
x2             0.9491      0.031     30.621      0.000         0.888     1.010
x3            -1.0498      0.204     -5.143      0.000        -1.450    -0.649
x4         -6.584e-05   1.49e-05     -4.434      0.000      -9.5e-05 -3.67e-05
x5             0.0003   7.41e-05      3.514      0.000         0.000     0.000
==============================================================================
Omnibus:                     1675.046   Durbin-Watson:                   1.965
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2573553.599
Skew:                          12.848   Prob(JB):                         0.00
Kurtosis:                     263.999   Cond. No.                     1.81e+07
==============================================================================
```

(b) OLS Regression Statistics for #patriots

From the table, we conclude **number of tweets, total number of retweets** and **sum of followers of the users posting the hashtag** to be significant features. For our regression model, R-squared value is **0.775** which means that our model is reasonable.

Linear model for #nfl

It can be seen that **number of tweets, total number of retweets** and **sum of followers of the users posting the hashtag** are the three most significant features. For our regression model, R-squared value is **0.568**. This could imply that linear model can explain data well.

Linear model for #patriots

Looking at the t-value and p-value, we can say **number of tweets, total number of retweets** and **sum of followers of the users posting the hashtag** to be significant features. For our regression model, R-squared value is **0.689**, which is good.

Linear model for #sb49

From the table, it can be said that **number of tweets, total number of retweets** and **maximum number of followers of the user posting the**

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.804
Model:                            OLS   Adj. R-squared:                  0.802
Method:                 Least Squares   F-statistic:                     437.5
Date:                Tue, 15 Mar 2016   Prob (F-statistic):         2.24e-186
Time:                        16:32:58   Log-Likelihood:                -5336.2
No. Observations:                 541   AIC:                         1.068e+04
Df Residuals:                     535   BIC:                         1.071e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        274.6222    402.826      0.682      0.496      -516.693  1065.937
x1             1.0902      0.103     10.593      0.000         0.888     1.292
x2            -0.1170      0.094     -1.238      0.216        -0.303     0.069
x3          3.647e-06     1.5e-05     0.243      0.808     -2.58e-05  3.31e-05
x4             0.0001   5.04e-05      1.989      0.047      1.25e-06     0.000
x5           -21.3409     29.723     -0.718      0.473       -79.730    37.048
==============================================================================
Omnibus:                     1076.891   Durbin-Watson:                   1.977
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1694081.886
Skew:                          13.912   Prob(JB):                         0.00
Kurtosis:                     275.725   Cond. No.                     1.83e+08
==============================================================================
```

(a) OLS Regression Statistics for #sb49

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.783
Model:                            OLS   Adj. R-squared:                  0.781
Method:                 Least Squares   F-statistic:                     437.5
Date:                Tue, 15 Mar 2016   Prob (F-statistic):         2.22e-198
Time:                        16:35:57   Log-Likelihood:                -6382.0
No. Observations:                 612   AIC:                         1.278e+04
Df Residuals:                     606   BIC:                         1.280e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const       -323.4217    674.014     -0.480      0.632     -1647.109  1000.265
x1             1.3405      0.297      4.517      0.000         0.758     1.923
x2             0.4013      0.147      2.725      0.007         0.112     0.690
x3            -0.0002   1.52e-05    -16.166      0.000        -0.000    -0.000
x4             0.0011      0.000      8.406      0.000         0.001     0.001
x5           -25.7477     48.632     -0.529      0.597      -121.256    69.760
==============================================================================
Omnibus:                     1014.193   Durbin-Watson:                   1.985
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1147772.306
Skew:                           9.506   Prob(JB):                         0.00
Kurtosis:                     214.304   Cond. No.                     2.42e+08
==============================================================================
```

(b) OLS Regression Statistics for #superbowl

hashtag to be significant features. We have a good training accuracy as our R-squared value is **0.804**, which is not very good.

Linear model for #superbowl

Looking at the t-value and p-value, we can say **number of tweets, total number of retweets** and **sum of the followers of the users posting the hashtag** to be signicant features. The training model is about 80% accurate, as the R-squared value is **0.783**.

3 Solution 3

There are several features that could affect the twitter popularity count. Out of these, we wanted to specifically hand-design features that were more user-dependent. We believe that user-dependent-features play a crucial rule in predicting the popularity/burst of a tweet and this is well explained in the reference paper[1].

Hence, we removed the features from section 2 that we deemed to be insignificant and in-place of those features, we added six new user-dependent-features: sum of friends count, sum of statuses count and sum of ranking score.

The reasons for why we chose these features are explained as follows:

Sum of friends count: In Twitter, you can become a friend to a person if you follow him/her and he/she follows you back. This was one of our crucial ideas behind selecting this as a feature. For example, if you are friends with one person on Twitter, then he/she is more likely to see your status and retweet it and vice versa. Since we believe that this aspect is crucial to making a hashtag popular, this was one of our features.

Sum of statuses count: Another feature we considered we was the sum of the statuses count of the users. A user with more statuses count is more likely to be an active user of Twitter and one with a lower status count is not likely to be very active. Hence, using this feature will help us predict the number of tweets in the next hour more accurately.

Sum of Ranking score: Ranking score indicates the importance of the user who is gener- ating the tweet. Higher the ranking score, the more in uential a user is and hence there is a greater probability of his tweets reaching a wider audience and making the hashtag popular. Hence this was naturally an additional feature for our prediction.

Number of hashtags: Number of hastags would be a significant feature because the probability of a user in twitter looking at a tweet would be higher if he notices more hashtags that he is interested in. For example, if there's a match between Superbowl and another team, and if the user is interested in that game, then there is a good possibility that the user is interested in that tweet as well.

Number of favorited tweets: The tweets which have a higher number for their "favorited" tweets would mostly have a higher probability of being noticed by other users while scrolling down. This is because, the user currently looking at the tweet would tend to look at it further when he thinks many other users like him (who have already looked at it) have favorited the tweet.

Number of citations: The number of citations for a given tweet could also matter as user's perspectives of the tweet will change accordingly.

Day of the week: We also believe day of the week matters to users as the tweets might gain popularity during particular days of the week when the matches are happening. If there are no matches during a particular day, then the popularity of a tweet posted on that day would be low.

3.1 Approach 1

From Figure 6, we find the top three features to be **sum of the followers of the users, sum of the ranking scores of the users** and the **sum of the friends count of the users** based on the p-value and the t-value measures (similar to section 2).

The plots below show the relation between each of these three features and the number of tweets in the next hour (predictant).

We tried to fit the model with 10 features as shown in Approach 2. However, the size of the files were very huge and therefore it was not easy to run on the code as combining all the 6 text files and running crashed multiple times on our computer (Specifications: i7, 8 Cores, 16 GB RAM 2.6 GHz).

Therefore, we tried to reduce the number of features from 10 to 7 and thereafter run or code. We fitted a linear regression model with the most significant features from section 2 and these three new features of our own adding to a total of 7 features. We label them x1, ..., x7 where

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.756
Model:                            OLS   Adj. R-squared:                  0.755
Method:                 Least Squares   F-statistic:                     1573.
Date:                Mon, 14 Mar 2016   Prob (F-statistic):               0.00
Time:                        01:03:11   Log-Likelihood:                -34849.
No. Observations:                3566   AIC:                         6.971e+04
Df Residuals:                    3558   BIC:                         6.976e+04
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const       -358.4976     79.674     -4.500      0.000      -514.708  -202.287
x1             8.6095      0.743     11.583      0.000         7.152    10.067
x2             0.6560      0.048     13.667      0.000         0.562     0.750
x3            -0.0001   6.24e-06    -21.290      0.000        -0.000    -0.000
x4             0.0003   3.55e-05      7.361      0.000         0.000     0.000
x5            -2.2758      0.192    -11.847      0.000        -2.652    -1.899
x6             0.0001   9.42e-06     13.201      0.000         0.000     0.000
x7             0.0004      0.000      2.694      0.007         0.000     0.001
==============================================================================
Omnibus:                     6271.991   Durbin-Watson:                   1.962
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         54124313.264
Skew:                          11.400   Prob(JB):                         0.00
Kurtosis:                     606.116   Cond. No.                     1.31e+08
==============================================================================
```

Figure 6: OLS Regression Statistics for features from Section 2 including hand-designed features

- x1 is sum of friends count
- x2 is sum of statuses count
- x3 is sum of Ranking score
- x4 is the number of tweets
- x5 is the total number of retweets
- x6 is the sum of the followers of the users posting the hashtag
- x7 is the maximum number of followers of the user posting the hashtag

Analysis

From the graphs, we can see that each of these features follow a linear relation with the number of tweets in the next hour. This is not surprising because one thing common with all these features is that they basically indicate the network of the user: followers of the user indicate how famous he, the ranking score of a user indicate how in influential he is, and the friends count of the user yet again indicates how many people follow him as friends. All these features highly correlate to the 'retweetability' of the tweet a user posts. Hence we can say that the features that we selected for our task are indeed good.

Training Accuracy

After we introduced these features, our training accuracy (indicated by the R-squared statistics) now increases to **0.756** (Figure 6).

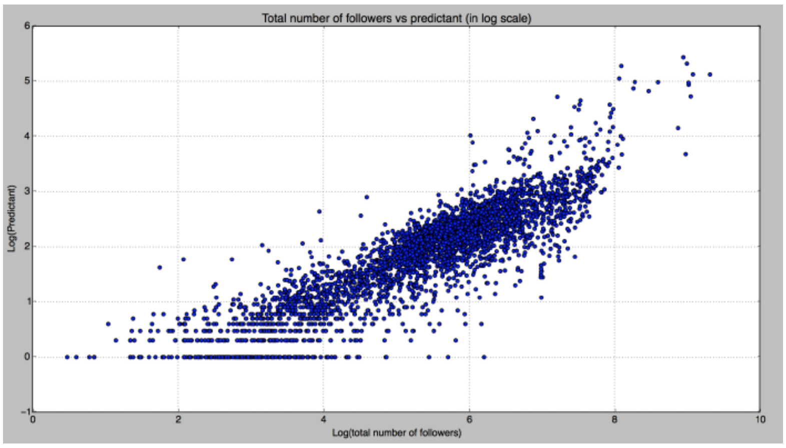


Figure 7: Scatter plot of predictant vs sum of number of followers of users

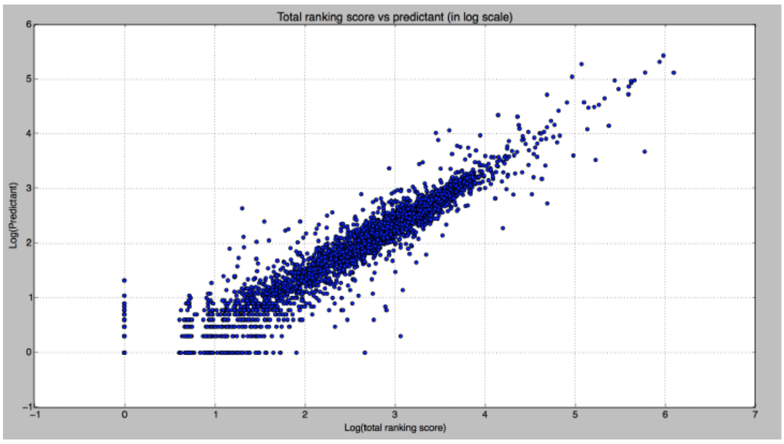


Figure 8: Scatter plot of predictant vs sum of ranking scores of users

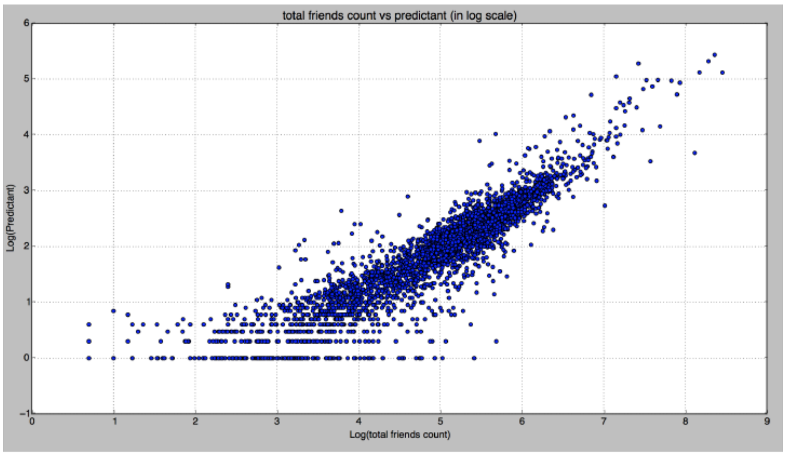


Figure 9: Scatter plot of predictant vs sum of friends counts of users

3.2 Approach 2

We fitted a linear regression model with the most significant features from section 2 and these three new features of our own adding to a total of 7 features. We label them x1, ..., x10 where

- x1 is the number of tweets
- x2 is the number of retweets
- x3 is the number of followers
- x4 is the number of hashtags
- x5 is the sum of favorites
- x6 is the sum of friends
- x7 is the number of verified users
- x8 is the sum of citations
- x9 is the sum of Ranking score
- x10 is the sum of statuses count

Linear Model for #gohawks
Looking at the t-value and p-value (we consider features with higher t-value magnitude and lower p-value to be significant), we can say **number of hashtags, followers, statuses count** are the 3 most significant features. For our regression model, after adding features, the accuracy has improved. This can be seen from increase in R-squared value to **0.622**.

Linear Model for #gopatriots

For this hashtag, **Sum of Hashtags, Number of tweets, and Sum of Ranking Score** are the 3 most significant features. We again see an improvement in Training accu- racy as compared to section 2. R-squared value is **0.875**.

Linear Model for #nfl

From the table, **Number of Retweets, Number of Hashtags, Number of Favorites** are the 3 most significant features. We again see an improvement in Training accuracy as compared to section 2. R-squared value is **0.748**.

Linear Model for #patriots

From the table, **Number of Retweets, Number of Hashtags, Number of Favorites** are the 3 most significant features. We again see an improvement in Training accuracy as compared to section 2. R-squared value is **0.658**.

Linear Model for #sb49

```
                          OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.622
Model:                            OLS   Adj. R-squared:                  0.618
Method:                 Least Squares   F-statistic:                     144.4
Date:                Tue, 21 Mar 2017   Prob (F-statistic):          8.85e-178
Time:                        01:31:03   Log-Likelihood:                -7048.0
No. Observations:                 889   AIC:                         1.412e+04
Df Residuals:                     878   BIC:                         1.417e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        -16.4004     25.062     -0.654      0.513       -65.589    32.788
x1            -8.4900      3.126     -2.716      0.007       -14.625    -2.355
x2            -0.0738      0.187     -0.394      0.694        -0.442     0.294
x3            -0.0003    4.6e-05     -6.500      0.000        -0.000    -0.000
x4             1.3220      0.233      5.664      0.000         0.864     1.780
x5             0.0603      0.083      0.729      0.466        -0.102     0.223
x6             0.0005      0.000      1.542      0.123        -0.000     0.001
x7            41.8756     10.089      4.150      0.000        22.073    61.678
x8            -0.0279      0.069     -0.404      0.686        -0.163     0.108
x9             1.1323      0.620      1.826      0.068        -0.085     2.349
x10            0.0001   2.61e-05      5.206      0.000      8.45e-05     0.000
==============================================================================
Omnibus:                     1892.129   Durbin-Watson:                   2.100
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          6511999.381
Skew:                          17.001   Prob(JB):                         0.00
Kurtosis:                     420.907   Cond. No.                     1.10e+07
==============================================================================
```

Figure 10: OLS Regression Statistics for #gohawks

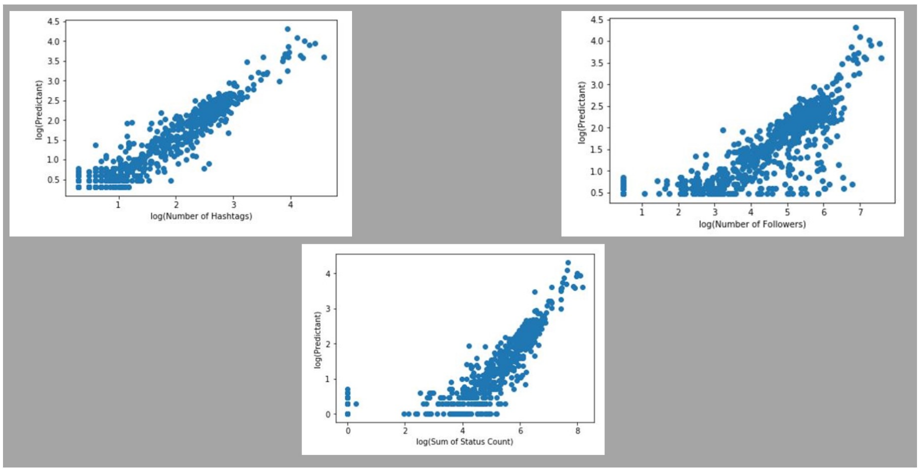


Figure 11: Relation between number of tweets in the next hour and 3 major features for #gohawks

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.875
Model:                            OLS   Adj. R-squared:                  0.873
Method:                 Least Squares   F-statistic:                     470.5
Date:                Mon, 20 Mar 2017   Prob (F-statistic):          1.10e-295
Time:                        23:32:25   Log-Likelihood:                -4110.0
No. Observations:                 683   AIC:                             8242.
Df Residuals:                     672   BIC:                             8292.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const          7.7072      3.950      1.951      0.051        -0.049    15.464
x1           -25.0801      1.882    -13.324      0.000       -28.776   -21.384
x2           -36.3220      3.775     -9.622      0.000       -43.734   -28.910
x3             0.0003   2.75e-05     10.553      0.000         0.000     0.000
x4             5.1811      0.262     19.743      0.000         4.666     5.696
x5            15.5616      3.368      4.620      0.000         8.948    22.175
x6            -0.0001      0.000     -0.596      0.551        -0.001     0.000
x7           -91.4846     13.695     -6.680      0.000      -118.375   -64.594
x8            -0.5486      0.154     -3.558      0.000        -0.851    -0.246
x9             3.6277      0.327     11.089      0.000         2.985     4.270
x10           -0.0001   1.82e-05     -7.427      0.000        -0.000 -9.93e-05
==============================================================================
Omnibus:                      548.088   Durbin-Watson:                   2.170
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            60301.359
Skew:                           2.847   Prob(JB):                         0.00
Kurtosis:                      48.678   Cond. No.                     1.00e+07
==============================================================================
```

Figure 12: OLS Regression Statistics for #gopartriots

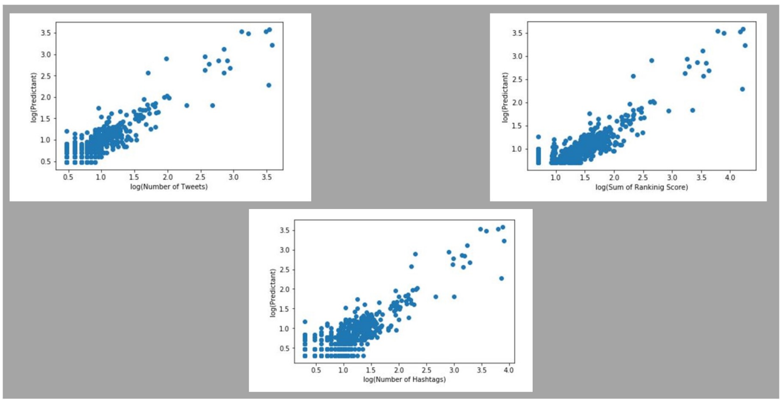


Figure 13: Relation between number of tweets in the next hour and 3 major features for #gopatriots

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.748
Model:                            OLS   Adj. R-squared:                  0.745
Method:                 Least Squares   F-statistic:                     263.3
Date:                Tue, 21 Mar 2017   Prob (F-statistic):         1.74e-257
Time:                        18:04:34   Log-Likelihood:                -8005.4
No. Observations:                 898   AIC:                         1.603e+04
Df Residuals:                     887   BIC:                         1.609e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        -83.2632     70.231     -1.186      0.236      -221.102    54.576
x1             5.9477      1.680      3.541      0.000         2.651     9.244
x2             0.8798      0.898      0.980      0.327        -0.882     2.641
x3            -0.0002   4.36e-05     -3.904      0.000        -0.000 -8.46e-05
x4             0.0965      0.156      0.620      0.535        -0.209     0.402
x5            -0.8856      0.791     -1.119      0.263        -2.439     0.668
x6            -0.0015      0.000     -4.419      0.000        -0.002    -0.001
x7            86.6335     11.688      7.412      0.000        63.695   109.572
x8            -0.4692      0.116     -4.031      0.000        -0.698    -0.241
x9            -1.2428      0.424     -2.928      0.004        -2.076    -0.410
x10            0.0001   1.36e-05      9.522      0.000         0.000     0.000
==============================================================================
Omnibus:                     1747.342   Durbin-Watson:                   1.723
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          3652944.295
Skew:                          13.993   Prob(JB):                         0.00
Kurtosis:                     314.200   Cond. No.                     4.13e+07
==============================================================================
```

Figure 14: OLS Regression Statistics for #nfl

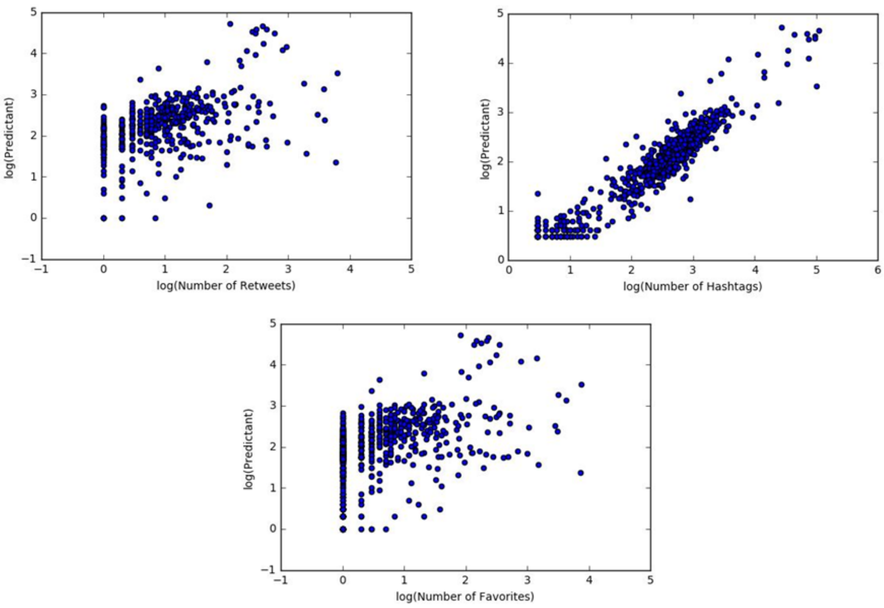


Figure 15: Relation between number of tweets in the next hour and 3 major features for #nfl

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.748
Model:                            OLS   Adj. R-squared:                  0.745
Method:                 Least Squares   F-statistic:                     263.3
Date:                Tue, 21 Mar 2017   Prob (F-statistic):         1.74e-257
Time:                        22:08:42   Log-Likelihood:                -8005.4
No. Observations:                 898   AIC:                         1.603e+04
Df Residuals:                     887   BIC:                         1.609e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        -83.2632     70.231     -1.186      0.236      -221.102    54.576
x1             5.9477      1.680      3.541      0.000         2.651     9.244
x2             0.8798      0.898      0.980      0.327        -0.882     2.641
x3            -0.0002   4.36e-05     -3.904      0.000        -0.000 -8.46e-05
x4             0.0965      0.156      0.620      0.535        -0.209     0.402
x5            -0.8856      0.791     -1.119      0.263        -2.439     0.668
x6            -0.0015      0.000     -4.419      0.000        -0.002    -0.001
x7            86.6335     11.688      7.412      0.000        63.695   109.572
x8            -0.4692      0.116     -4.031      0.000        -0.698    -0.241
x9            -1.2428      0.424     -2.928      0.004        -2.076    -0.410
x10            0.0001   1.36e-05      9.522      0.000         0.000     0.000
==============================================================================
Omnibus:                     1747.342   Durbin-Watson:                   1.723
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          3652944.296
Skew:                          13.993   Prob(JB):                         0.00
Kurtosis:                     314.200   Cond. No.                     4.13e+07
==============================================================================
```

Figure 16: OLS Regression Statistics for #patriots

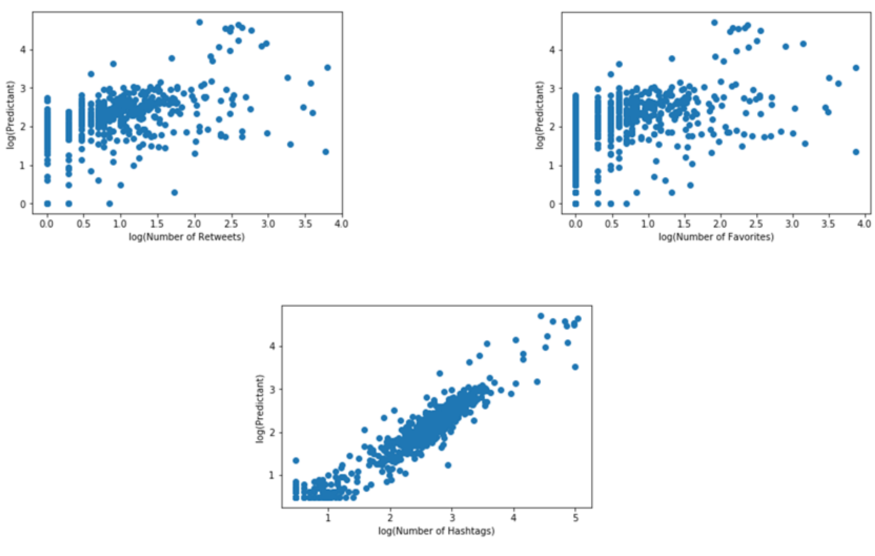


Figure 17: Relation between number of tweets in the next hour and 3 major features for #patriots

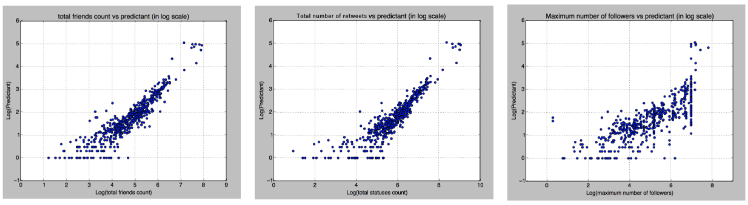


Figure 18: Relation between number of tweets in the next hour and 3 major features for #sb49

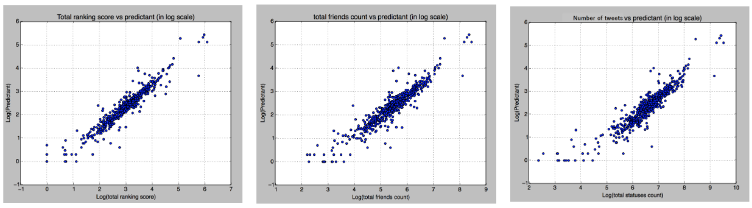


Figure 19: Relation between number of tweets in the next hour and 3 major features for #superbowl

From the table, **sum of friends count, the sum of the followers of the users posting the hashtag and number of re-tweets** are the 3 most significant features. We again see an improvement in Training accu- racy as compared to section 2. R-squared value is **0.825**.

Linear Model for #superbowl

From the table, **sum of Ranking score, sum of the followers of the users posting the hashtag and number of tweets** are the 3 most significant features. We again see an improvement in Training accuracy as compared to section 2. R-squared value is **0.822**.

4 Solution 4

In this section, instead of running a general regression model over the entire period, we run a regression model for three specic time periods:

- Before Feb. 1, 8:00 a.m
- Between Feb. 1, 8:00 a.m. and 8:00 p.m
- After Feb. 1, 8:00 p.m

Important Observation

One important observation to note here is that, Feb. 1 is the day on which the Superbowl took place and hence our hypothesis is that this time period should contain the maximum burst of the tweets. To validates the same, we considered two approaches and tabulated them.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.724
Model:                            OLS   Adj. R-squared:                  0.694
Method:                 Least Squares   F-statistic:                     23.99
Date:                Mon, 14 Mar 2016   Prob (F-statistic):           1.17e-15
Time:                        01:39:58   Log-Likelihood:                -838.07
No. Observations:                  72   AIC:                             1692.
Df Residuals:                      64   BIC:                             1710.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const       1917.7492   4274.300      0.449      0.655     -6621.146  1.05e+04
x1             8.0375      6.795      1.183      0.241        -5.538    21.613
x2             1.1856      0.429      2.762      0.007         0.328     2.043
x3            -0.0002   5.78e-05     -3.813      0.000        -0.000    -0.000
x4             0.0009      0.001      1.717      0.091        -0.000     0.002
x5            -2.5008      1.750     -1.429      0.158        -5.997     0.996
x6             0.0003   9.71e-05      3.275      0.002         0.000     0.001
x7            -0.0010      0.001     -0.719      0.475        -0.004     0.002
==============================================================================
Omnibus:                       53.283   Durbin-Watson:                   2.185
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              349.348
Skew:                           1.999   Prob(JB):                     1.38e-76
Kurtosis:                      13.023   Cond. No.                     1.02e+09
==============================================================================
```

Figure 20: OLS Regression Statistics for model trained on period1

4.1 Approach 1

Cross-Validation

We first split the data into three specific time periods (before Feb.1, 8:00 a.m, between Feb.1, 8:00 a.m. and 8:00 p.m, after Feb.1, 8:00 p.m) based on the time stamp (one hour bin) of the tweet. Hence, we now have three sets of training data each for a specific period.

We now train a regression model for each one of these periods. We split each data into train-test split and perform a 10-fold cross-validation. We divide the training data into ten subsets and for each of the ten folds, we t a model on 9 parts of the data and test on the nal part. We then average our errors $|N_{predicted} - N_{real}|$ over the ten folds.

Our **Mean Absolute Errors** errors for each of the three periods:

- Before Feb.1, 8:00 a.m: **136.81**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **21832.47**
- After Feb.1, 8:00 p.m.: **86.27**

Analysis of Results

From the results above we can see that for the time period before and after the Superbowl the errors are very less. This is because there are relatively lesser number of tweets over a longer time period for **Before Feb.1, 8:00 a.m** and **After Feb.1, 8:00 p.m.**

OLS Regression Results

Dep. Variable:	y	R-squared:	0.479
Model:	OLS	Adj. R-squared:	0.478
Method:	Least Squares	F-statistic:	366.3
Date:	Mon, 14 Mar 2016	Prob (F-statistic):	0.00
Time:	01:39:58	Log-Likelihood:	-21720.
No. Observations:	2796	AIC:	4.346e+04
Df Residuals:	2788	BIC:	4.350e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
const	47.7833	12.793	3.735	0.000	22.698	72.868
x1	-2.9857	0.568	-5.260	0.000	-4.099	-1.873
x2	-0.0140	0.017	-0.818	0.414	-0.047	0.020
x3	-1.703e-07	3.71e-06	-0.046	0.963	-7.45e-06	7.11e-06
x4	-1.432e-06	1.14e-05	-0.126	0.900	-2.38e-05	2.09e-05
x5	0.6719	0.124	5.418	0.000	0.429	0.915
x6	6.725e-06	2.59e-06	2.594	0.010	1.64e-06	1.18e-05
x7	0.0005	5.84e-05	9.115	0.000	0.000	0.001

Omnibus:	5679.625	Durbin-Watson:	1.933
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21821573.866
Skew:	16.357	Prob(JB):	0.00
Kurtosis:	434.555	Cond. No.	1.56e+07

Figure 21: OLS Regression Statistics for model trained on period2

OLS Regression Results

Dep. Variable:	y	R-squared:	0.899
Model:	OLS	Adj. R-squared:	0.898
Method:	Least Squares	F-statistic:	878.0
Date:	Mon, 14 Mar 2016	Prob (F-statistic):	0.00
Time:	01:39:58	Log-Likelihood:	-4643.9
No. Observations:	698	AIC:	9304.
Df Residuals:	690	BIC:	9340.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
const	17.9780	9.329	1.927	0.054	-0.339	36.295
x1	-2.9497	0.420	-7.029	0.000	-3.774	-2.126
x2	0.0030	0.005	0.573	0.567	-0.007	0.013
x3	-3.823e-06	2.01e-06	-1.903	0.057	-7.77e-06	1.21e-07
x4	1.576e-05	4.38e-06	3.597	0.000	7.16e-06	2.44e-05
x5	0.8940	0.096	9.348	0.000	0.706	1.082
x6	-1.382e-06	1.28e-06	-1.081	0.280	-3.89e-06	1.13e-06
x7	-6.596e-05	3.5e-05	-1.883	0.060	-0.000	2.8e-06

Omnibus:	924.260	Durbin-Watson:	1.960
Prob(Omnibus):	0.000	Jarque-Bera (JB):	351300.606
Skew:	6.446	Prob(JB):	0.00
Kurtosis:	112.146	Cond. No.	2.66e+07

Figure 22: OLS Regression Statistics for model trained on period3

However for the time period during the **Superbowl: Between Feb.1, 8:00 a.m. and 8:00 p.m.**, there is a huge burst of tweets. Also, the number of training examples that we have for this period is very less compared to the other two periods as we are taking one-hour bins and the time period is just 12 hours. Hence, our model is prone to high errors during this time.

Probable Solution

A solution could be to use **sliding windows** to increase the number of data points. By using a sliding window, it is possible to increase the number of training examples, thereby reducing the possibility of errors.

If we had indeed had a smaller time-period, say 10 minute bins instead of 1 hour bins, we anticipate that our model would have performed better since we would have a larger number of examples which might have lead to a better accuracy.

4.2 Approach 2

Here, we do cross-validation training and testing for each of the six linear models separately. Like the previous approach, we adopt a 10-fold cross validation for each linear model. The results of our cross-validation training and test are as follows:

Linear Model for #gohawks

- Before Feb.1, 8:00 a.m: **266.17**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **43954.250**
- After Feb.1, 8:00 p.m.: **25.89**

Linear Model for #gopatriots

- Before Feb.1, 8:00 a.m: **18.64**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **5487.45**
- After Feb.1, 8:00 p.m.: **3.51**

Linear Model for #nfl

- Before Feb.1, 8:00 a.m: **107.03**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **11100.45**
- After Feb.1, 8:00 p.m.: **115.45**

Linear Model for #patriots

- Before Feb.1, 8:00 a.m: **205.78**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **29456.02**

- After Feb.1, 8:00 p.m.: **70.89**

Linear Model for #sb49

- Before Feb.1, 8:00 a.m: **48.104**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **62594.86**
- After Feb.1, 8:00 p.m.: **104.23**

Linear Model for #superbowl

- Before Feb.1, 8:00 a.m: **210.842**
- Between Feb.1, 8:00 a.m. and 8:00 p.m: **95027.14**
- After Feb.1, 8:00 p.m.: **178.21**

Analysis of Results

From the results above we can see that for the time period before and after the Superbowl the errors are very less. This is because there are relatively lesser number of tweets over a longer time period for **Before Feb.1, 8:00 a.m** and **After Feb.1, 8:00 p.m.**

However for the time period during the **Superbowl: Between Feb.1, 8:00 a.m. and 8:00 p.m.**, there is a huge burst of tweets. Also, the number of training examples that we have for this period is very less compared to the other two periods as we are taking one-hour bins and the time period is just 12 hours. Hence, our model is prone to high errors during this time.

Probable Solution

A solution could be to use **sliding windows** to increase the number of data points. By using a sliding window, it is possible to increase the number of training examples, thereby reducing the possibility of errors.

If we had indeed had a smaller time-period, say 10 minute bins instead of 1 hour bins, we anticipate that our model would have performed better since we would have a larger number of examples which might have lead to a better accuracy.

5 Solution 5

For this problem, we are given test data which contain 10 windows of 6 hour time periods: sample1-period1.txt, sample2-period2.txt, sample3-period3.txt, sample4-period1.txt, sample5-period1.txt, sample6-period2.txt, sample7-period3.txt, sample8-period1.txt, sample9- period2.txt, sample10-period3.txt.

in order to make predictions for the number of tweets in the next hour for each of these six-hour windows. We can leverage that fact that we have built three different regression models for each interval: Before Feb.1, 8:00 a.m (period 1), Between Feb.1, 8:00 a.m. and 8:00 p.m (period 2), After Feb.1,8:00 p.m. (period 3).

Also, we know from the test data as to which period each sample belongs to. Hence, we segregate the test data into three sets each belonging to period 1, period 2 and period 3.

For example, if we want to test for sample1-period1.txt, we will send it to the model trained on tweets during period 1 i.e Before Feb.1, 8:00 a.m. If we want to test for sample6- period2.txt, then we would be using the model trained on tweets during period 2 i.e Between Feb.1, 8:00 a.m. and 8:00 p.m.

Hence, in this problem, for each test case, we send it to one of the fitted models, and obtain the prediction of number of tweets for the next hour for each of them.

The results for number of tweets predicted in the next hour (using the features of the last hour of each window) for each sample window is reported below.

Test case belonging to period 1:

- sample1-period1.txt: **198.5262**
- sample4-period1.txt: **182.705**
- sample5-period1.txt: **230.6620**
- sample8-period1.txt: **6.3584**

Test case belonging to period 2:

- sample2-period2.txt: **233380.7584**
- sample6-period2.txt: **26066.8325**
- sample9-period2.txt: **1943.213**

Test case belonging to period 3:

- sample3-period3.txt: **426.1041**
- sample7-period3.txt: **38.5555**
- sample10-period3.txt: **59.1661**

Observations

- Before Superbowl (during period 1), the predicted number of tweets is very less. This makes sense because the number of tweets before Superbowl would be less as

- During period 2 (exactly during the period of Superbowl), predicted number of tweets shoots to thousands or hundreds of thousands. This is expected as the number of tweets generated will naturally be higher during this period as it is the Superbowl period and therefore the popularity is high.
- After the Superbowl (during period 3), number of predicted tweets reduces again. one can explain this by the fact that the excitement of Superbowl must have gone down and so is the number of tweets.

6 Solution 6

Given only the textual content of the tweet, we first segregated the tweets based on the location. Location in this case is a type of metadata. After classifying the tweet data based on metadata (which is location in our case), we could observe better results.

Importance of Metadata

In most information technology usages, the prefix of meta conveys "an underlying definition or description. More precisely, however, metadata describes data containing specific information like type, length, textual description and other characteristics. **In this particular case**, we used **Location** as a metadata feature and classified the tweets further. This is the method of implementation:

Tweets containing the following substrings in the location field were taken for "Washington". They were:

"Seattle","Seattle, WA" , "WA", "Washington", "Seattle, Washington", "Kirkland, WA", "Kirkland, Washington".

Washington DC was excluded from the list.

Tweets containing the following substrings in the location field were taken for "Massachusetts". They were:

"Boston", "Worcester", "Boston, MA", "MA", "Massachusetts".

The ROC curve and confusion matrix are as shown below for the case of SVM as shown in Figure 23.

Accuracy: **0.7322**

Precision: **0.75**

Recall: **0.73**

Observations

Accuracy increased considerably in the case of applying SVM alone. Location is a decent metadata when applied with the SVM classifier, as SVM is basically a **Maximum Margin Classifier**. In other words, it minimizes the

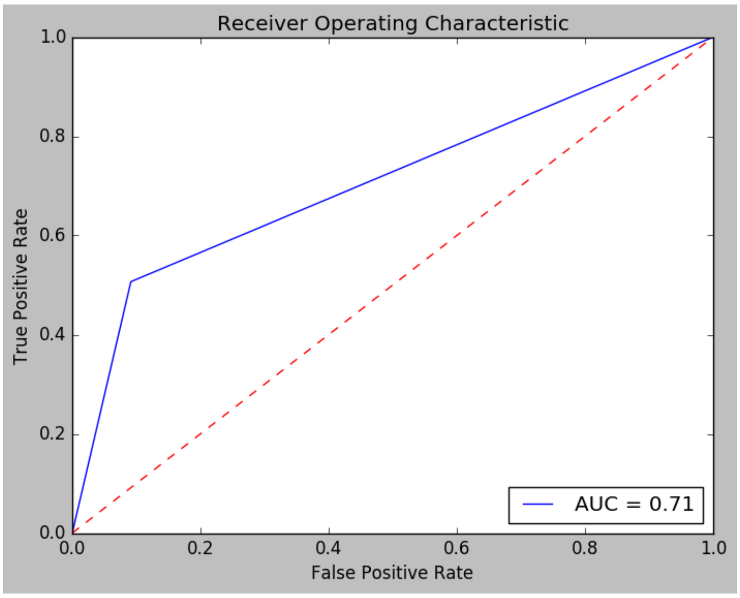


Figure 23: ROC Curve for Superbowl with SVM

Table 2: Confusion Matrix for #Superbowl with SVM

	Predicted Washington	Predicted Massachusetts
Actual Washington	12221	1230
Actual Massachusetts	5186	5329

Table 3: Confusion Matrix for #Superbowl with Soft Margin SVM

	Predicted Washington	Predicted Massachusetts
Actual Washington	12225	896
Actual Massachusetts	5749	4766

Table 4: Confusion Matrix for #Superbowl with Logistic Regression

	Predicted Washington	Predicted Massachusetts
Actual Washington	12201	1250
Actual Massachusetts	5189	5326

```
Best Gamma Value: 100
===========soft_margin_SVM===========
              precision    recall  f1-score   support

   Washington       0.69      0.93      0.79     13451
Massachusetts       0.84      0.45      0.59     10515

  avg / total       0.75      0.72      0.70     23966

accuracy: 0.722732203956
Confusion Matrix
[[12555   896]
 [ 5749  4766]]
```

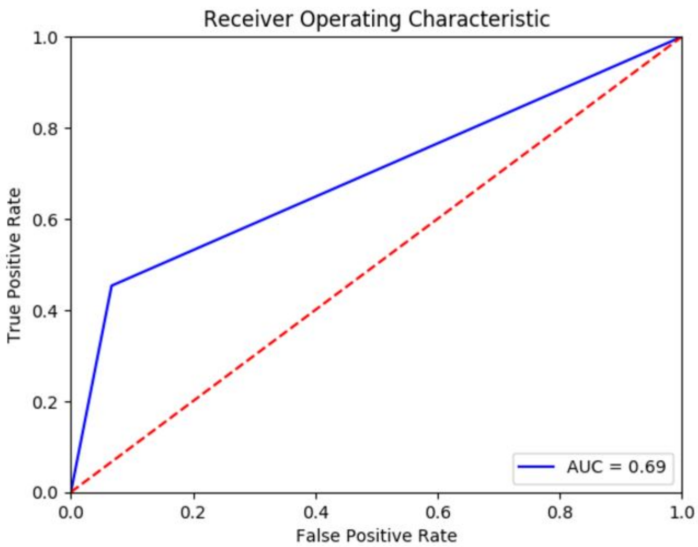


Figure 24: ROC Curve, Accuracy, Precision & Confusion Matrix for Soft Margin SVM

```
===========Logistic Regularization(Unregularized)===========
               precision    recall  f1-score   support

   Washington       0.70      0.91      0.79     13451
Massachusetts       0.81      0.51      0.62     10515

  avg / total       0.75      0.73      0.72     23966

accuracy: 0.731327714262
Confusion Matrix
[[12201  1250]
 [ 5189  5326]]
```

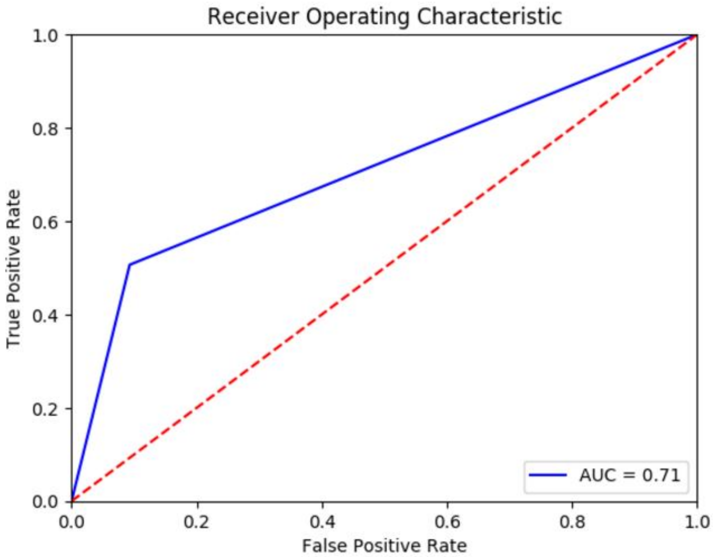


Figure 25: ROC Curve, Accuracy, Precision & Confusion Matrix for Logistic Regression

error for the closest points on both sides of the margin. The results we obtained were not great for location based classification, however, The different classification algorithms that we tried are:

- k-nearest neighbours
- Support Vector Machine (SVM)
- Logistic Regression
- Hybrid Classifiers (SVM + Logistic Regression)
- Decision Trees
- Soft Margin SVM
- Random Forest

Among these techniques, the two most powerful techniques which we could find in terms of results were **Hybrid Classifiers**, **Soft Margin SVM**, **Logistic Regression** and **SVM**.

The **Boosting Algorithm** proposes to generate several learning sets iteratively, and in a given iteration object which is classified incorrectly in the previous iteration should be drawn with a higher probability. The final decision is done on the basis of weighted voting rule. However, in our case, we noticed that it was better if we used SVM alone over a Hybrid technique in terms of results. We anticipate the reason for the same due to the following reasons:

6.1 Comparison of Results for different Classification Algorithms

Table 5: Accuracy, Precision and Recall comparison for different methods

	Accuracy	Precision	Recall
Logistic Regression	0.7313	0.75	0.73
SVM	**0.7322**	**0.75**	**0.73**
Soft Margin SVM	0.7222	0.75	0.72

Advantages of SVM

Firstly it has a regularisation parameter, which makes the user think about avoiding over-fitting. Secondly it uses the kernel trick, so you can build in expert knowledge about the problem via engineering the kernel. Thirdly an SVM is defined by a convex optimisation problem (no local minima) for which there are efficient methods (e.g. SMO). Lastly, it is an approximation to a bound on the test error rate, and there is a substantial body of theory behind it which suggests it should be a good idea.

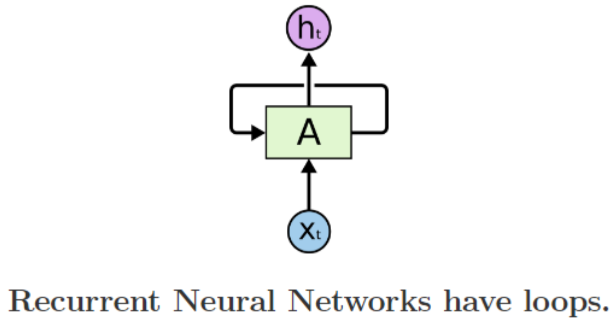


Figure 26: RNN with Loops

7 Solution 7

Automatic Tweet Generation on a Trending Topic using Recurrent Neural Networks

Preview of Results

- We try to use deep-learning technique to generate tweets automatically, very similar to the tweet data that we used to train the recurrent neural network.
- Following this, we **Interpret the Data** we thereby obtained from the automatically generated tweets and complete a Qualitative and Quantitative Analysis

Recurrent Neural Networks

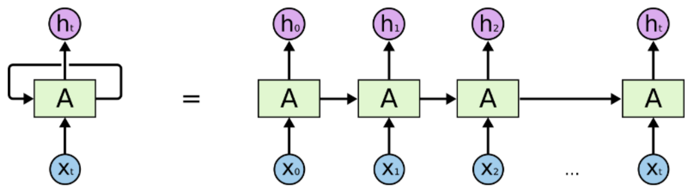
Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence.

Traditional neural networks can't do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.

In Figure 26, a chunk of neural network, A, looks at some input x_t and outputs a value h_t. A loop allows information to be passed from one step of the network to the next.

These loops make recurrent neural networks seem kind of mysterious. However, if you think a bit more, it turns out that they aren't all that different



An unrolled recurrent neural network.

Figure 27: Uncontrolled RNN

than a normal neural network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop as shown in Figure 27.

This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists. They're the natural architecture of neural network to use for such data.

In order to define our own problem statement and execute it using the Twitter data set, we tried to experiment with the recurrent neural network (RNN) architecture for a specific task and analyse how it performs on the Twitter dataset. Following the same, we try to interpret the data by collecting similar/different features and thereby plot graphs to identify its relations with other features.

For our task, we attempt to build a character level language model specific to each hashtag and perform 'tweet hallucination' (automatic generation of "tweet like text" by the algorithm) using RNN.

In other words, our model takes one text file as an input and trains a recurrent neural network that learns to predict the next character in a sequence. The neural network can then be used to generate text on a character by character basis that will look like the original training data (in our case, the twitter data). The RNN models the probability distribution of the next character in a sequence from the previous outputs in the sequence which allows us to generate one character at a time. During the test time, a 'seed' character is placed into the input and it tries to predict the next character in the sequence based on the probability distribution obtained from our model. We then try to predict successive characters and this process goes on [2][3].

Pre-processing of the text data

Since we want the algorithm to generate twitter-like text on its own, we need to clean the twitter data set that contains a lot of meta data and send in only the relevant information.In our case, this is the 'description' attribute of the tweets. For our task, we are considering only the English tweets, so we set the 'language' attribute to 'EN (english)' and extract only the tweets in order and store them in a text le. We perform the same for all the hashtags individually and the cleaned les which we will be giving as input data to our RNN algorithm are cleaned gohawks.txt, cleaned gopatriots.txt, cleaned n .txt, cleaned superbowl.txt.

Software and Implementation Details

Although there were many tool kits available for RNN in Torch, keras etc, we used the toolkit provided in rnnlm as they provided a good documentation and were easy to implement. For our implementation, we used a **16GB Ram 8 core i7 CPU running at 2.6 GHz** and on an average it took around 4200 seconds for training the RNN up to 100000 time steps.

Experiments performed

In our task, we want to analyze both qualitatively and quantitatively the effectiveness of recurrent neural networks for text hallucination for our Twitter data set.

Firstly, during training we wanted to analyze the set of hyper parameters on the convergence of our algorithm. There were many hyper parameters that we tuned and tested our model on.

The first hyper parameter was the number of units in the hidden layer of the RNN. Since for a number of hidden units, the algorithm takes a long time to compute (more than 2 hours), we restricted ourselves to hidden units of sizes 100 and 256 for our analysis. The second hyper-parameter that we considered was the learning rate. For this we experimented with learning rates of 0.01, 0.1, 1, 10. For our analysis of learning rates, we fixed our number of hidden units in RNN as 100.

Finally, we wanted to do a qualitative analysis of how well the model performed. Hence, we also perform a comparison of the sample ground truth twitter text data and the text generated by the model after 100, 1000, 10000 and 100000 iterations for each of the hashtags. Although there are more state-of-the-art sophisticated models with many hidden layers and LSTM units, due to computational and time restrictions, we restricted our analysis to a very minimalistic model. But even in this minimalistic approach, our model is able to perform really well and identifies the important recurring patterns in the dataset

Qualitative Analysis

```
#GOHAWKS file
Ground Truth:
all the way from Maui, Kalua's ready for the game tomorrow.
@KING5Seattle @Seahawks #WeAre12 #GoHawks #twelfies
http://t.co/F9wk6yWZWj
@Seahawks GAME DAY! #GoHawks #ImIn http://t.co/nEP9MJzwHn
@chadwmiller all good!
We are rooting for the hawks tonight!
#gohawks #GOKNIGHTS @RealMikeRob NICE SOCKS .
```

```
After 1000 iterations
 \xf #E\B9f\x99OETf\xYOE
P"@90\x9Gf\x99f8fd0 #!\e tse\x8bx0"\-\x9\B2\x9\xf\x90\x9f#x"f0
bG Bsl fric2G\)8\x98 #Gobritn honds OJ
bf0eopg_ciouy_S9 f\xgonHc\x@9\"x8f\x9\x9\d9f\dSf4@ff\x9xo8 Zf l-

After 100000 iterations
MOT Hawks that Fame!!? \xf0\x9f\x92\x99 #gohawks
its #ers stermes ster sonfing thy ni. Champions Sustran.
@serecarrroH wicken WAN1 Championshy Happs #GoHawks'
Nee Keer at just onte. #GoHawks #GoHaw
```

```
#GOPATRIOTS file
Ground Truth:
Today is the day. Go to the Super Bowl or go home.
More than a game. It's a way of life.http://t.co/WcImgNuWMp
You just wouldn't understand. #GoPackGo #GoPatriots

After 1000 iterations
gatawks athe. h tn tpat vro- than1 \xo? woep: Tht.
1Tve tt mea @Je at.c4/t in #NI'
be2s vam Su2. le iseehis.cucLt. 7Sl'
whttp:ik veaoline Botking #GoHawksqOu
yoviprvenlkN\xfs\x9f.'2veo to frtnytycke

After 100000 iterations
Chint at best!!! Of- @kitbinno @wAlBchgame
Plown the #GoPatriots http://t.co/Ovr7veKXwuK"
uttit! #GoPatriots #SupCCheamiust Now altt. @PatriotsWIN.
```

```
#SUPERBOWL file
Ground Truth:
NFL News: Super Bowl 2015:
AFC, NFC Conference Championship Predictions and Vegas Odds
http://t.co/RceKmSM9mB #Football #NFL #SuperBowl

After 1000 iterations
aoaponriLExx4Bt hTf'ntt2/oiatherBpofldiKhalo-B
bulr ::o tt2#Sel #updRALc #oorancc! PoV ys ctactpedix
berialil'ca" ukamlilQ coIX 9nO
bes f hhZLe !! thSim #Su@sferIXld.es #wusn ootire #Suporakipt

After 100000 iterations
HawkV #Patriots. #Brady the #viperyighin win happY
```

```
@PatriotsNetion @Superbowl having to canch @Coltoon matColts
#SuperBowl49 #Seahawks happoon a See
```

```
#SB49 file
Ground Truth:
I wore my @PatMcAfeeShow shirt, my new Colts warmup last Sunday
I will be wearing same thing Today!
#GoColts #BoomStick #SB49 #Lombardi

After 1000 iterations
igi.Wlcktesliod'seo/mt'X\A/
xIXLl Eac #DuwlCs @bwig fesafwome mlrX2bU@2q
athricVty jvr TAsBol ys p :CkVx";
y'Hagon fet #Susis sLL @DS)kg  jooises @1on DA!x//x@f f Crn
/ONOto 5sssol hluzleg woowIs'LIjG

After 100000 iterations
 uperBowlXLIX #Tondy. Axe'.  #NewTVS Super!BoTr ut News
 @Reexficits yeah ne. #PatriotsNatiovs"
"Nevs! 1 mood in the. #SuperBowlXLIX'
Seabalay -I's AnmedBinut its rebN! #Seahawks lokiuveuss Colt ther
```

```
#NFL file
Ground Truth:
Your one stop destination for All Things NFL.
Rules, results, stats, news, pictures, info more.
NFL fans please follow #nfl #USA #football

After 1000 iterations
ROB3TATHIBCUl3 #NFLDref\xa2\x86 http//ttco/ChMPFSNI
XLNFW #NFL'Cer #Rape #nfl'
Proatthit Howgs\xe2\x80\xafc #Ra6
Bfs #Tfl #Walt ##NFL\x90\xhQgWl #NFL'
"Rapn: Hape #1JKS SA6


After 100000 iterations
#Sonfake this moupto funt, pame is #Seahawks #NFL
#NFLPlayoffs Han see gereres]
#Seattle ero a #Seahawks #Patsle oy\n#gement that whit #Seahawks
```

We display the text generated by the RNN after 1000 and 100000 iterations. We can clearly notice that after 100, text is mostly junk and there is no meaning

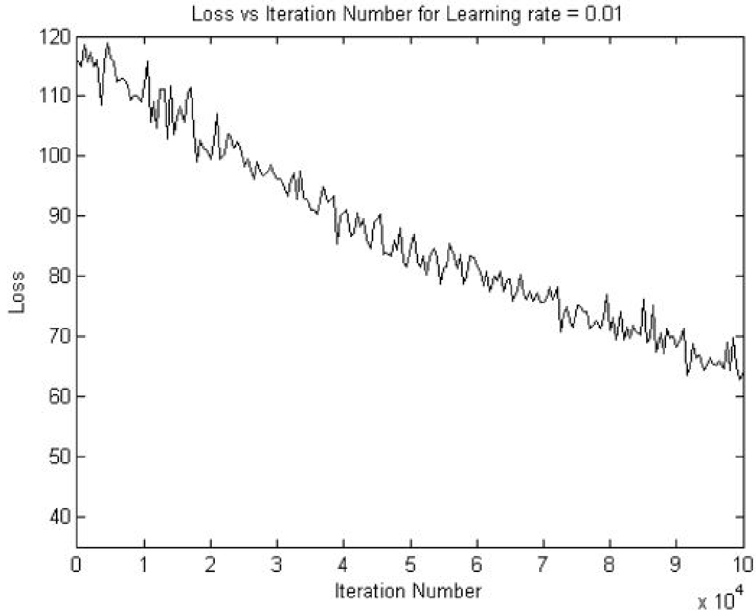


Figure 28: loss function vs Iteration number,learning rate 0.01

to it. But as the number of iterations increase, the text generated by the RNN seems to be more and more similar to our original tweets. And it can be seen that although most of the text is still gibberish, the algorithm is able to hallucinate the most important recurring words in the data set in a cohesive manner which depicts the power of the Recurring Neural Network.

Quantitative Analysis

The performance of the algorithm with the variations in the learning rate and hidden unit dimensions is explored in this section. As a specific example, we have plotted the graphs for change in the learning rate vs the number of time steps for different values of learning rate = 0.01, 0.1, 1 (Figures 28, 29, 30) for the hashtag **#gohawks**.

We can see that for very low learning rates such as 0.01, the algorithm converges very slowly. However for a learning rate of 0.1, the algorithm converges in optimal time. But for very high learning rates such as 1, the algorithm shoots up and diverges and it fails to find the global minima.

Tweet Interpretation

It is very interesting to derive metrics out of tweets generated automatically by the RNN. One of the first metric we derived is the number of hash tags present in tweets generated every hundredth iteration. From Figure 31, it can be seen that there was a maximum of 12 hash tags at one point.

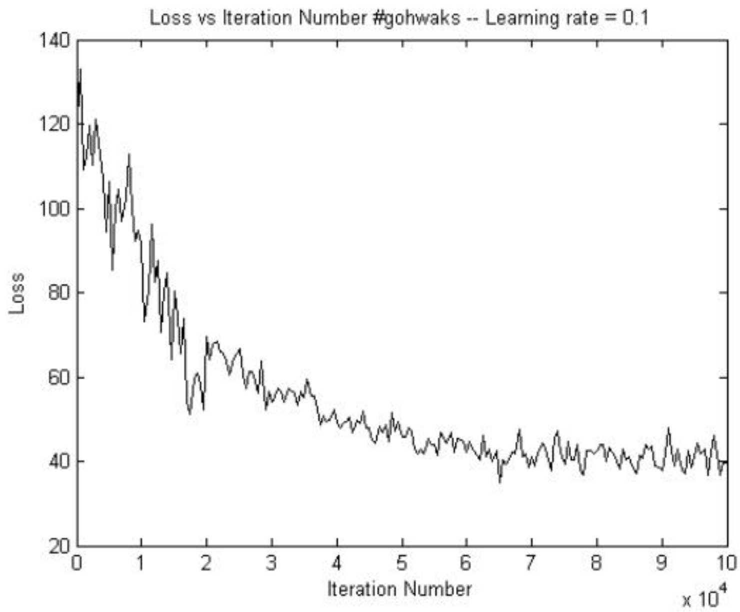


Figure 29: loss function vs Iteration number,learning rate 0.1

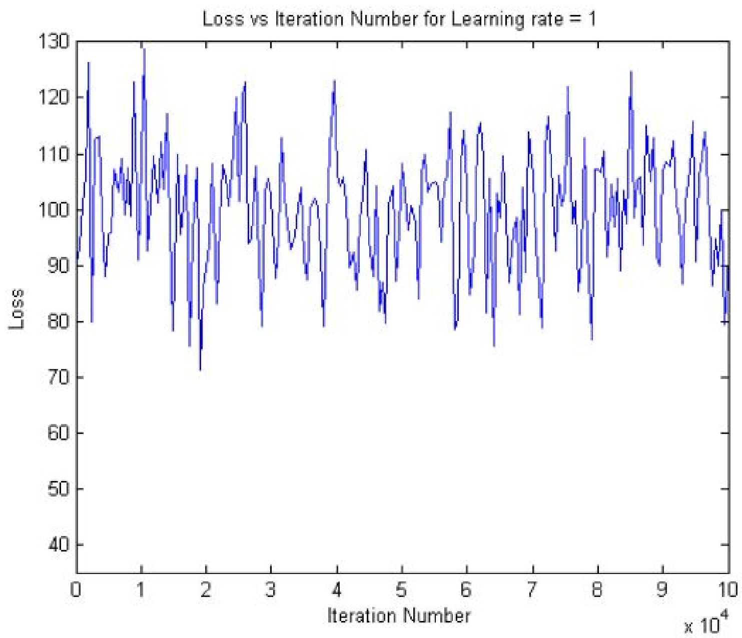


Figure 30: loss function vs Iteration number,learning rate 1

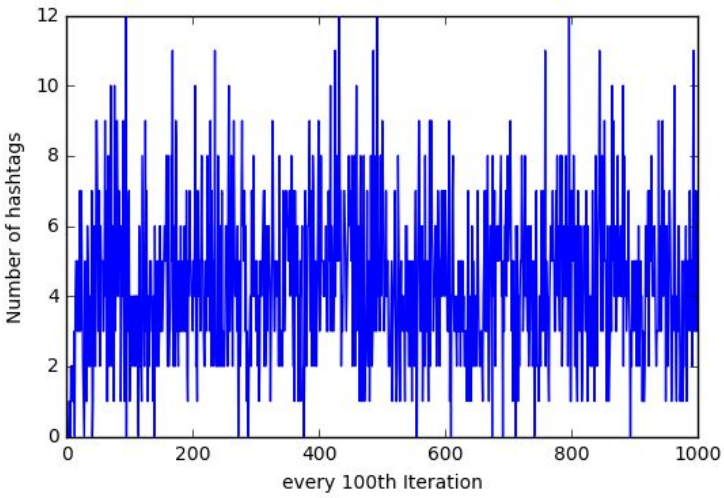


Figure 31: Iteration vs number of hash tags

Next we measured the number of valid English words present in the tweets generated. The scatter plot in Figure 32 shows that there are about 15 to 20 valid English words.

The hash tags #nfl, #patriots,#gohawks, #superbowl and #sb49 are all closely related to the main hash tag #gopatriots. So we measured the number of occurrences of #nfl, #patriots,#gohawks, #superbowl and #sb49 in the tweets at each iteration. Figure 33 shows that maximum of 3 of those hash tags were generated by RNN.

Conclusion

In this section, we worked on the task of text hallucination using RNN and quantitatively and qualitatively analyzed the effectiveness of RNN for this specific. We also investigated the quality of the said generated tweets to understand the real performance of this method.

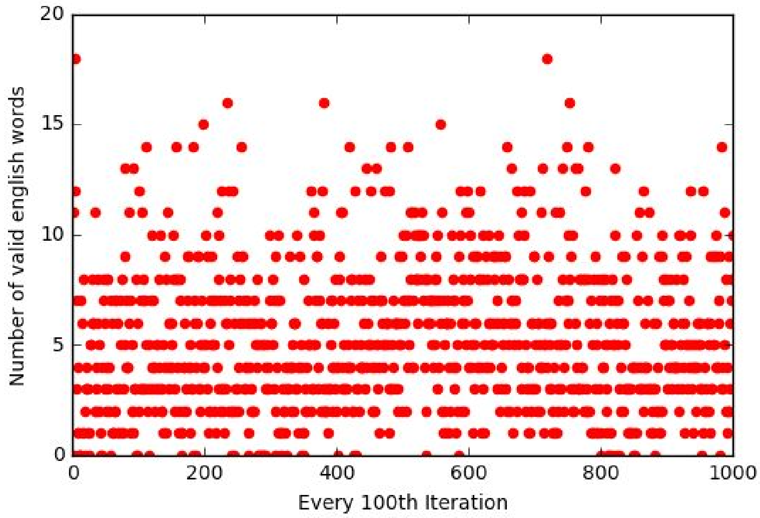


Figure 32: Iteration vs number of hash tags

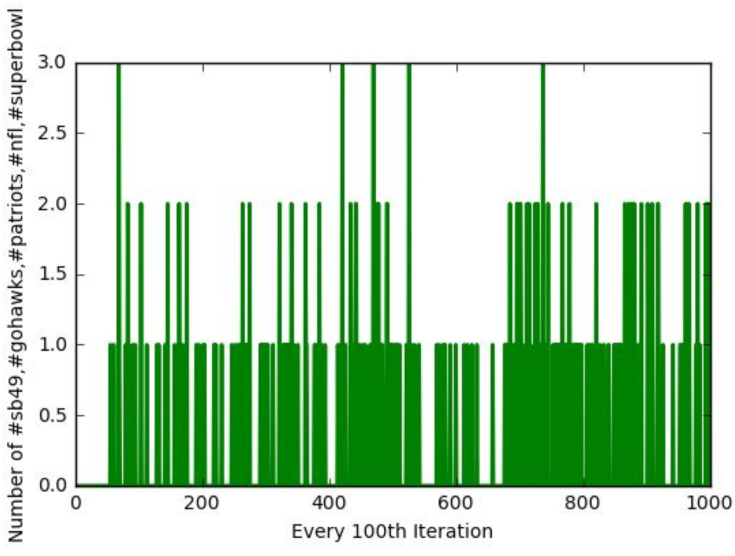


Figure 33: Iteration vs number of #nfl, #patriots,#gohawks, #superbowl and #sb49

References

[1] On the Real-time Prediction Problems of Bursting Hashtags in Twitter,
http://arxiv.org/pdf/1401.2018v2.pdf

[2] Long Term Short Memory,
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] Recurrent Neural Networks,
http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/