Fantastic Beasts: The Secrets of Dumbledore Measuring Promotional Effectiveness and User Responses Using Twitter API

Phuong Nguyen
Qiutong Shi
Xiaoyang Zhou
Keke Qin
Xiaocheng Guo

Johns Hopkins University
Carey Business School
Social Media Analytics, Spring 2022
Professor Tao Chen

Executive Summary

"Fantastic Beast: The Secrets of Dumbledore," released on April 15th 2022, faced declining box office sales in the first week of its release. We hope to investigate how users respond to the movie promotions as well as promotional effectiveness before and after the movie release period. We utilized regression analysis, sentiment analysis, content analysis, and correlation analysis. We found that 1) both user sharing links and not sharing links have negative attitudes when mentioning the Harry Potter series, 2) users have negative attitude toward the replacement of actors by Warner company, and 3) users who share links tend to have a negative attitude toward the news that the LGBT dialogue was removed in China. From the non-user perspective, we propose four recommendations: First, develop marketing materials independently from the Harry Potter franchise. Second, work directly with non-user accounts such as film reviewers, news sites, and movie superfans. Third, limit link usage in sharing promotional content. Fourth, include hashtags and photos.

1.0 Intro

On April 15th, 2022, a new movie named "Fantastic Beasts: The Secrets of Dumbledore" was released. It is the third episode of the Fantastic Beasts series which are subsequent movies of Harry Potter. Since the numerous Harry Potter fans, Fantastic Beasts becomes a trending topic across multiple social media platforms every time a related movie is released. However, the audience' preferences towards the Fantastic Beasts series is gradually declining. The first Fantastic Beasts movie generated the highest box office value of $814 and an IMDb rating of 7.2, while the second series generated $650 billion and IMDb rating of 6.5.

Since the Fantastic Beasts series is currently facing the obstacles of decreasing box office and IMDB ratings, the objective of the research is digging into audience responses to the most recently released Fantastic Beasts Movie.

2.0 Research questions

We have two research questions. First, we would like to investigate how users respond to the movie promotion and release. Then, we are interested in the promotional effectiveness of the Fantastic Beasts. In the end, we will provide some recommendations that the company can use to increase engagement from users and thus enhance promotional effectiveness.

3.0 Methodology and Results

3.1 Data Process and Assumption

We retrieved data using two keywords, the full name of the movie “Fantastic Beasts: The Secrets of Dumbledore” and the short from “Fantastic Beasts”. Both datasets cover tweets from April 5th to April 30th (around 1 week before and 2 weeks after the movie release). These two keywords were chosen under the assumption that official accounts would talk about the movie using the full movie name, while users discussing the movie will refer to it in its shortened form: Fantastic Beast.

We differentiated non-users (ex. Official accounts or other promotional accounts) and users in the dataset based on the source of tweets and the keywords. Non-users tend to use the full name, while real people generally use Twitter on a mobile device (including iphone, ipad, android and web app) and refer to the movie as “Fantastic Beasts”. We also further define tweets containing external links posted by movie promoters who are excited about the film, thus we also isolated them from real users.

After the above data cleaning, there are 28312 real users, 3755 promoters and 23182 non-users who posted 37473, 11656 and 51362 tweets respectively.

3.2 Bass Model

The bass model for users and non-users shares a similar pattern. The bass model expects that the word of mouth reaches a peak around April 13th to 14th (the day the movie was released). However, the reality is not as predicted. The number of tweets reached the first peak on April 7th (Thursday, a week before the movie was released) and decreased for one to two days. At this stage, real users talked more about the movie and the decrease is more significant for them, even though it was a Friday. The “Fantastic BeastS” is tweeted most frequently two days before the movie was released and the heat dramatically dropped on April 13th. Especially for non-users, the fluctuation was more intense, the assumption is that official accounts and influencers put the most effort on promoting the movie in advance and they stopped posting tweets the day before the movie was released. As the movie was released on April 14th, the word of mouth mounted another peak. After the movie was released, word of mouth decreased as expected, but there is a small jump on April 23rd. The promotional messages triggered a larger buzz for non-user groups.

3.3 Sentiment Analysis and Word Clouds

3.3.1 Sentiment Analysis

We conducted sentiment analysis for the complete user dataset, and repeated the same process for users with and without external links to see if there is any result that could support our initial assumption that there is differentiation among these two groups of users and other meaningful outcomes.

For the complete user dataset, the average and mean sentiment score is around 0.17, showing a rather neutral attitude. The maximum sentiment score is 2.01, and the minimum is -1.2247. It seems that people who enjoy this movie do enjoy this movie a lot. For users who share contents with external links, the average and mean sentiment score is around 0.15, with the maximum sentiment score being 1.39 and minimum being -0.92. The range of sentiment score from this group of users is smaller than that of the whole group. For users who do not share contents with external links, the average and mean sentiment score is around 0.2, which is slightly more positive than that of the whole group and the other user group respectively. This group of users also have the more extreme sentiment scores in both the positive and negative direction (Appendix 3).

We then calculated the daily average sentiment score of users sharing and not sharing any links and compared them side by side. The time scope is April 5th to April 30th, covering the release dates. The trend is rather consistent with average daily sentiment scores get lower as time progresses. Additionally, users who do not share an external link have higher average sentiment scores before the movie release date and during the first five days of the movie's initial launch. We also observe two peaks from users who do not share any links. These peaks appear on April 8th, the movie's initial release date in China, and April 15th, the release date in other countries. We speculate that people who do not include any links have stronger attitudes toward the movie, and are more likely to be fans.

3.3.2 Word Clouds

In the preprocessing of the textual data, we remove stop words such as "Fantastic", "Beasts", "Secrets" and other words related to the title of the movie to get a more precise understanding of what words are brought up by users (Appendix 4). We notice that for both users sharing and not sharing external links, the words mentioned are rather consistent. This includes words related to the actors who have been involved in scandals, the company Warner, the Harry Potter franchise, and news related to the movie. We try to quantify these effects and discover more meaningful relationships in our topic model section.

3.4.1 Topic Model

We use topic models to see what aspects of the movie users engage the most. Our basic assumption for this part is that if the text contains one or more Url, we believe that the tweet has external links. In this way we divided the data into two parts(tweets with external links and tweets without external links) to see if there's any difference between the users who share links and the users who do not share links.

We generate five topic models with each topic containing ten words. For users who use external links, the topic that they discuss the most is "HP", which is composed mainly of Harry Potter series' key words. It is reasonable that people care about this topic because most of the fans who watch Fantastic Beasts are probably Harry Potter's fans or at least watched the movie

before. They also mention the performance of the movie frequently, which is also reasonable since people who share external links are mostly official or promotion accounts according to our assumption, and they would be more willing to predict what prizes the movie will win.

For users who don't use external links, they discuss the action the most. According to the key words we generated from our data, it seems that people are excited about the release of the movie and they can't wait to watch it. Other than that, people also mention Harry Potter related topics quite often, which is the same as the users who share external links.

3.4.2 Topic Model & Review Correlation

Next, we will dive deep into people's attitudes toward Fantastic Beasts. We generate correlation scores for each topic model and review's sentimental scores, and the negative score represents negative attitude, vice versa. We have three main findings in general.

1. Users are unsatisfied with the replacement of actors by Warner company.
2. Both users sharing links and not sharing links have a negative attitude when mentioning the Harry Potter series.
3. Users who share links have opposite attitudes towards the performance of the movie and the box office of the movie.

First, the correlation between the topic actor and reviews' sentimental score is -0.23189, which means people have a negative attitude towards this topic. The reason for this is obvious: users are mad at the replacement of actors by Warner company. It is widely known that Johnny Depp, who was the original actor of the role Grindelwald, was replaced by another actor Mads Mikkelsen due to the scandal reported recently. Although many people believe that he was innocent, the Warner company still asked him to resign from the movie. This breaking news resulted in a lengthy defense from his fans and even J.K Rowling, the creator of Harry Potter world, and many people are still angry about the decision made by the company, and thus have a negative attitude towards the company and the replacement of Johnny.

Second, the correlation for tweets with external links is -0.02196 and without external links is -0.07839, which means that both users sharing links and not sharing links tend to have a negative attitude when mentioning the Harry Potter series. After reading some of the related tweets, we found that many people are disappointed about Fantastic Beasts and believe that this franchise is not as good as the other Harry Potter movies. Some of them feel that the whole franchise is a fraud because the whole movie has no relationship with Harry Potter's wizarding world except for the same background setting. Moreover, many of them agree that the main storyline of Fantastic Beasts is not catching their eyes and the plots are not creative enough. Other than that, the words "Harry" and "Potter" are usually combined with "Johnny", which means that people's attitude towards Johnny's leaving might affect the overall score for this topic and decrease it further.

Last but not the least, we found that the correlation score for performance is 0.05404, which is positive. However, the correlation score for box tickets is -0.01456, which is negative. It

is weird because if people are positive about the performance of the movie, they should also have higher expectations for the box office. There are two possible reasons. First, the sample size might affect the final score. The topic proportion score for performance is around 0.249 and for box office is around 0.172, which might cause some bias and thus lead to the opposite outcome. Second, the people who predict the performance of the movie might be different from the people who predict the box office. Because these two topics are for tweets with external links, there are many official and promotion accounts included, and they would probably make predictions about the possible prizes that movie could win. However, the real users who reshare the links might care more about the box office. Since they reshare the link, based on the assumption we made earlier, these people should be the promoters for the movie, and thus will be more likely to have a positive attitude toward the box office.

3.4.3 Engagement Analysis

After identifying the user's responses to the movie, we look to do some exploratory analysis as well as regression analysis to see how promotional accounts can better improve their promotional efforts for the movie in the future. Plotting the frequency of tweets from non-user accounts from April 4th to May 2nd **(**See Appendix 5**)**, we found a similar spike to the user side in the number of tweets during the release date of the movie. However, what is interesting is that the number of retweets actually spiked 3 days before the spike in the number of non-user tweets. Assuming that the number of retweets represents the number of engagement from users, we can see that users are already engaging with promotional content from non-user accounts much earlier than when promotional content peaks. This can be explained by a few factors. First, it could be that the majority of engagements came from a few accounts that have early access to the movie. Second, users get more excited about the movie prior to seeing it and are retweeting promotional contents more frequently to share their excitement.

To see what non-users are posting their content from, we charted the top 12 most popular sources that non-users publish their content from (Appendix 5). Besides Twitter for iPhone and Twitter for Android, we see a high number of tweets being published from social media management tools such as Buffer (1.77%), dlvr.it (5.09%), Hootsuite Inc. (3.44%), Twitter Media Studio (1.65%), and Wordpress (5.24%). Most engagement came from @WatchmenID, @DEADLINE, @RegalMovies, @TheKimCromwell, @MauraLeamy, @HIDEO_KOJIMA_EN, @FantasticBeasts and @rameshlaus.

We conduct further analysis on the hashtag used by non-user accounts. We generate a frequency table for the most used hashtags (see Appendix 5). The most used hashtags are #SecretsofDumbledore and #FantasticBeasts. Interestingly, the hashtags with the highest engagements are related to Johnny Depp and Harry Potter (see Appendix 5). Identifying whether or not using hashtags increase the number of likes and retweets, we conduct regression analysis to identify correlation between including hashtags and number of likes and retweets. For number of likes (see Appendix 5), at alpha level = 0.001 (99% confidence level), we can conclude that including hashtags correlates with an 16.837 increase in likes on average for non-user accounts.

Similarly, for the number of retweets (see Appendix 5), at alpha level = 0.001 (99% confidence level), we can conclude that including hashtags correlates with a 3.5188 increase on average for non-users accounts.

We conduct a similar analysis for the inclusion of media and links in a promotional post. For media inclusion, we found that 35% of non-user content includes media (see Appendix 5). We conduct regression analysis for the relationship between media inclusion and engagement.At alpha-level = 0.001, we can conclude with 99% confidence that including media (a photo in our case) to a tweet correlates with a 45.537 increase in likes and 7.1897 in number of retweets. We repeated the same process for link inclusion and found that about 66.36% of non-user content includes a link (see Appendix 5). We found that at alpha-level = 0.05, we can conclude at 95% confidence that link inclusion correlates with a 10.179 decrease in number of likes (see Appendix). However, there is no evidence to conclude that link inclusion has any significant effect on the number of retweets.

4.0. Recommendations

Based on our analysis, we found four significant recommendations for promoters/marketing team of the Fantastic Beast franchise. First, develop marketing materials independently from the Harry Potter franchise. Second, work directly with non-user accounts such as film reviewers, news sites, and movie superfans. Third, limit link usage in sharing promotional content. Fourth, include hashtags and photos.

Firstly, based on our analysis of the user's response to the movie, we found that the majority of negative sentiments correlate with topics relating to the Harry Potter franchise, the replacement of actor Johnny Depp by Warner, and the removal of the LGBTQ+ dialogue in the movie release in China. For the second and third associations, these are events that are difficult for the promotional team to control, thus we will not be paying close attention to them. For the first negative association, however, the promotional team for Fantastic Beast could consider a new creative direction for the franchise to be a magical spin-off of the original beloved series rather than a complicated expansion of the universe that offers little satisfaction to the fans. Consider *Parks and Recs*, a popular spinoff of the award-winning *The Office*. The two shows are developed within the same universe, using the same style of storytelling, yet are independently well-received by audiences.

Secondly, from our analysis, we found that non-user accounts such as film reviewers, news sites, and movie superfans heavily sway public opinions about the film even before its release. We found that fans were already engaging with contents from these sites and talking about the movies' casting, trailers, and storyline way before they even saw it in theaters. It would be advisable for the team to identify key influencers in the network and offer them incentives to connect with the audience.

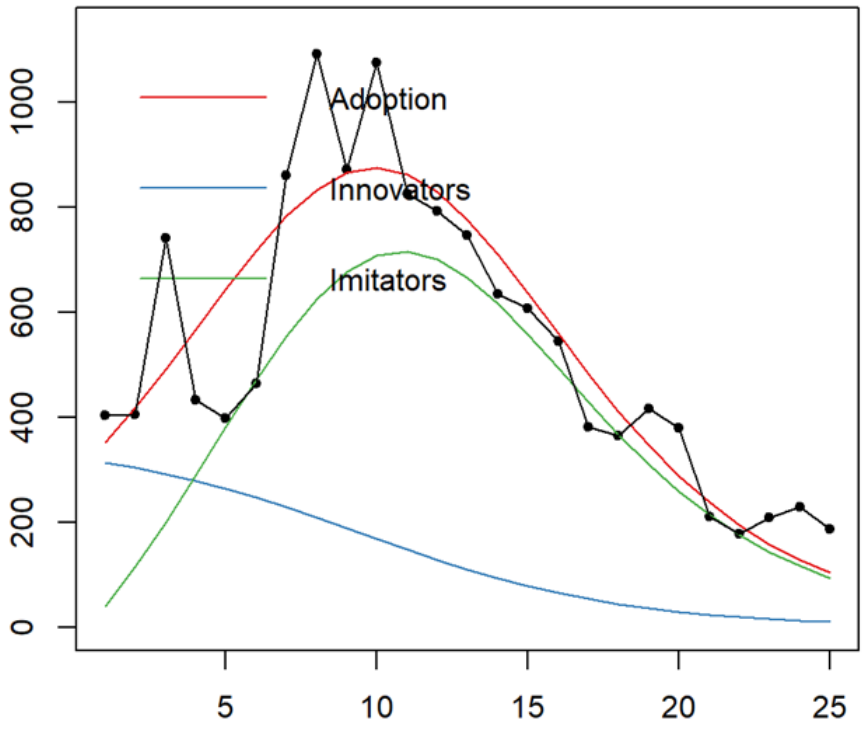
Thirdly, limit link usage in sharing promotional content. As mentioned above, link inclusion correlates with a 10.179 decrease in number of likes. Whether or not this decrease is significant for the promotional team is up for interpretation, however, we offer a few theories why link inclusion is bad in sharing social media posts. There are a few possible explanations for

this. 1) Links often take up a large majority of the content, and for a character-restricted platform such as Twitter, this significantly reduces the conversation that non-users can have with their audiences, 2) The Twitter algorithm might favor tweets with no links over tweets with link, and 3) Links are often associated with a Call-to-Action, which creates expectations for users to take a further step in their engagement with the content.

Finally, we found statistically significant results that including links and hashtags correlates with a higher overall engagement in number of likes and number of retweets.

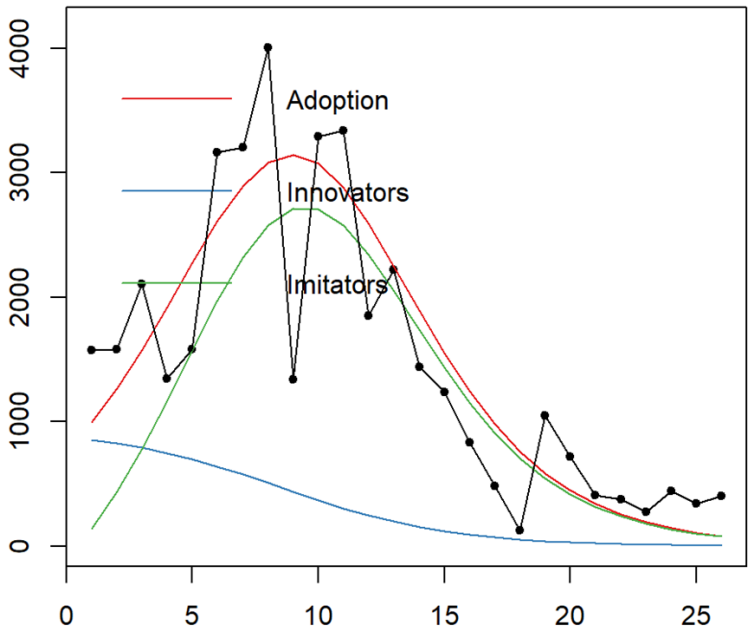
Appendix 1 Bass model for real users

```
> fit
bass model

Parameters:
                              Estimate p-value
p - Coefficient of innovation     0.0235      NA
q - Coefficient of imitation      0.2066      NA
m - Market potential          13680.0679      NA

sigma: 123.2248
```



Appendix 2 Bass model for Non Users

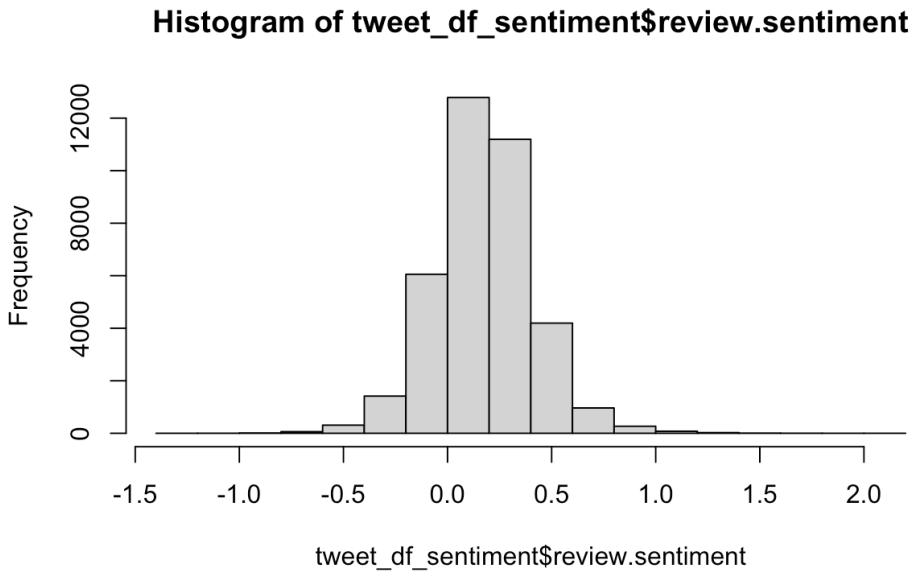
```
> fit
bass model

Parameters:
                                 Estimate p-value
p - Coefficient of innovation      0.0223      NA
q - Coefficient of imitation       0.2742      NA
m - Market potential           39309.5511      NA

sigma: 571.9022
```



Appendix 3 Sentiment Analysis Appendix

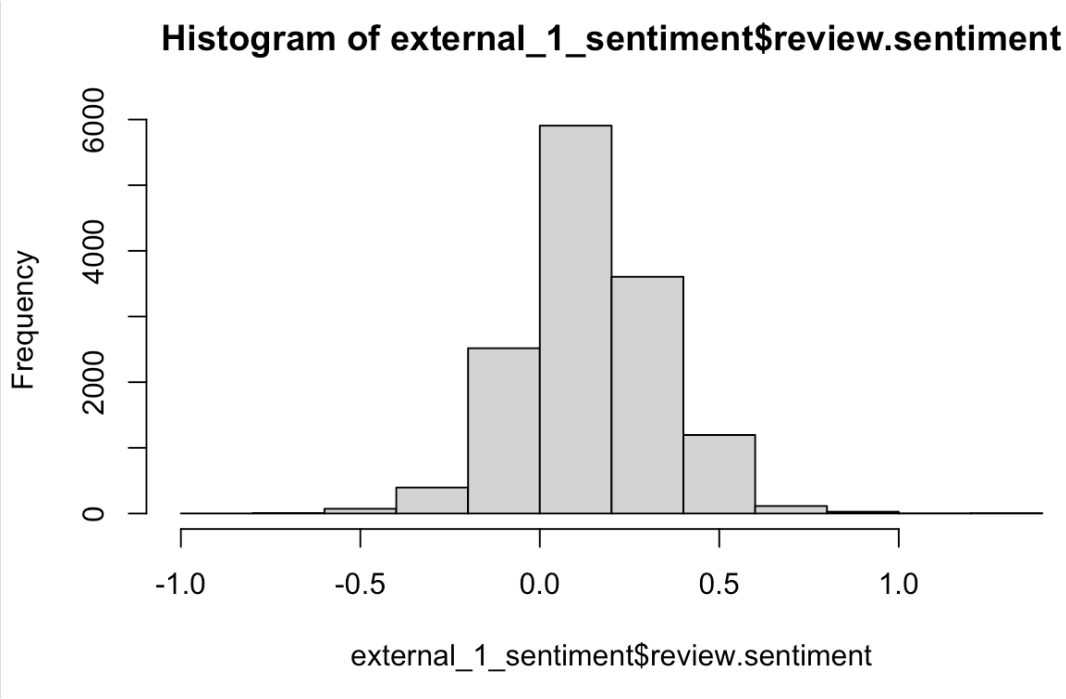
Sentiment Analysis for whole data set

```
> summary(tweet_df_sentiment$review.sentiment)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.2247  0.0416  0.1768  0.1784  0.3062  2.0100
```

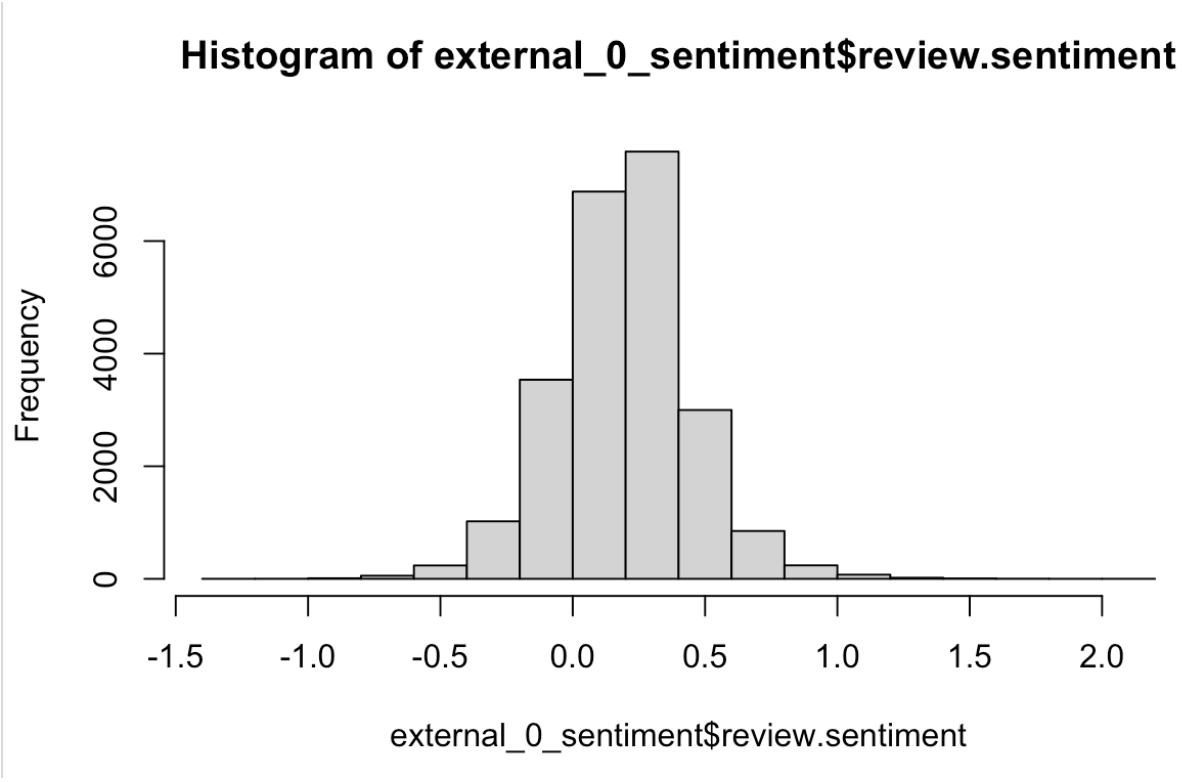


Sentiment Analysis for users sharing external links

```
> summary(external_1_sentiment$review.sentiment)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.92197  0.03494  0.15000  0.14858  0.23797  1.38911
```

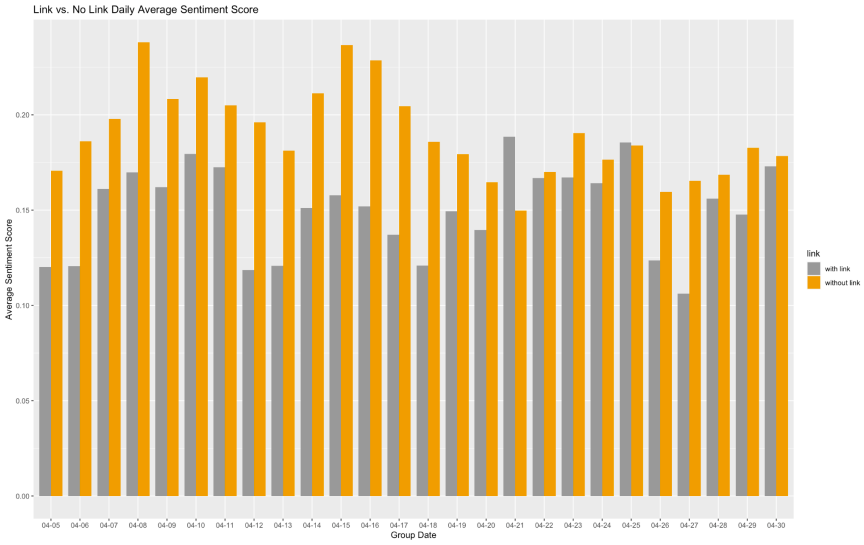


Sentiment Analysis for users not sharing external links



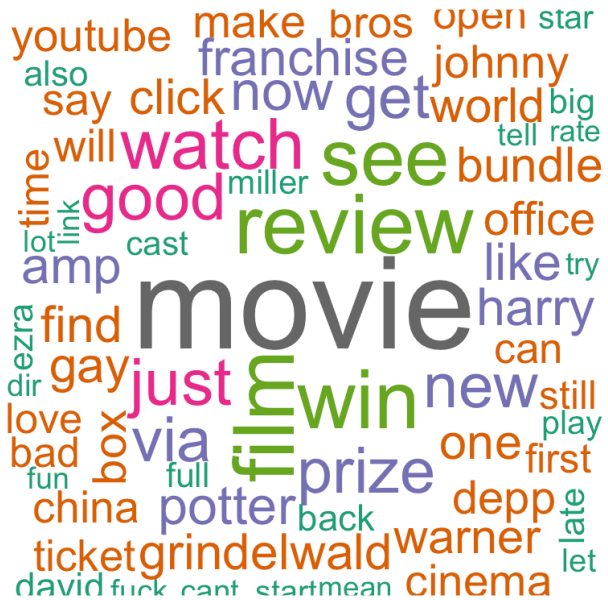

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.22474  0.04467  0.20035  0.19590  0.33541  2.01000
```

Plot the daily average sentiment score against one another

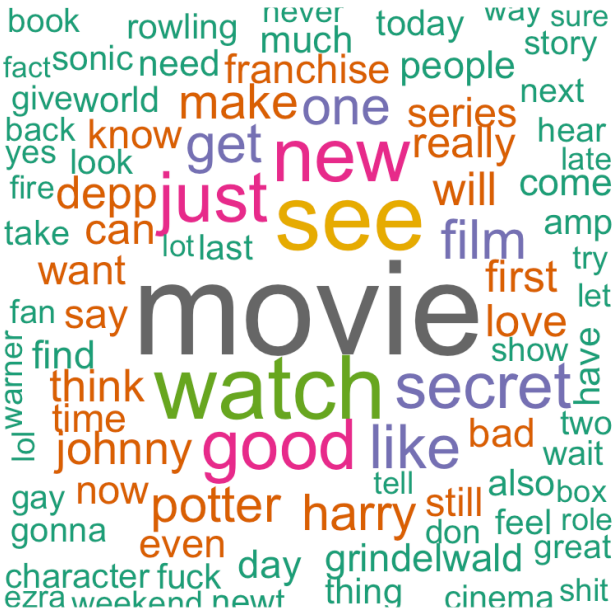


Word Clouds

With external links



Without external links



Appendix Topic model for real users (without external line)

```
> TOPIC = top.topic.words(result$topics, 10, by.score=TRUE)
> TOPIC
      [,1]          [,2]          [,3]         [,4]     [,5]
 [1,] "face"        "dollar"      "watch"      "johnny" "potter"
 [2,] "dumbledore"  "dumbledore"  "see"        "depp"   "harry"
 [3,] "secret"      "secret"      "new"        "warner" "like"
 [4,] "cry"         "sonic"       "movie"      "fire"   "series"
 [5,] "loudly"      "box"         "one"        "hear"   "franchise"
 [6,] "smile"       "office"      "fantastic"  "amber"  "newt"
 [7,] "heart"       "day"         "today"      "ezra"   "story"
 [8,] "tear"        "law"         "wanna"      "people" "world"
 [9,] "fantastic"   "jude"       "first"      "pirate" "make"
[10,] "joy"         "morbius"     "beast"      "miller" "stick"
```

```
> topicproportion[1:10,]
           [,1] [,2]      [,3]      [,4]      [,5]
 [1,] 1.0000000 0.00 0.0000000 0.0000000 0.0000000
 [2,] 0.0000000 0.00 0.0000000 0.7500000 0.2500000
 [3,] 0.0000000 0.00 0.5833333 0.0000000 0.4166667
 [4,] 0.0000000 0.30 0.6000000 0.1000000 0.0000000
 [5,] 0.2272727 0.00 0.2272727 0.2727273 0.2727273
 [6,] 0.5000000 0.00 0.5000000 0.0000000 0.0000000
 [7,] 0.2500000 0.00 0.7500000 0.0000000 0.0000000
 [8,] 0.4400000 0.00 0.0000000 0.0400000 0.5200000
 [9,] 0.1000000 0.00 0.0000000 0.0000000 0.9000000
[10,] 0.0000000 0.04 0.0400000 0.0000000 0.9200000
> colMeans(topicproportion)
[1] 0.1848823 0.1321290 0.2518232 0.1902802 0.2408854
```

Appendix Topic model for Promoters (with external line)

```
> TOPIC4
      [,1]          [,2]          [,3]       [,4]       [,5]
 [1,] "heart"       "secret"      "dollar"   "face"     "good"
 [2,] "star"        "dumbledore"  "box"      "gay"      "franchise"
 [3,] "white"       "prize"       "office"   "warner"   "like"
 [4,] "medium"      "review"      "ticket"   "bros"     "still"
 [5,] "newt"        "click"       "weekend"  "china"    "potter"
 [6,] "secret"      "bundle"      "sonic"    "remove"   "depp"
 [7,] "dumbledore"  "win"         "open"     "dialogue" "johnny"
 [8,] "eddie"       "fantastic"   "secret"   "smile"    "see"
 [9,] "point"       "beast"       "now"      "chinese"  "one"
[10,] "full"        "pack"        "cinema"   "censor"   "harry"
```

```
> topicproportion4[1:10,]
           [,1]       [,2]       [,3]       [,4]       [,5]
 [1,] 0.6666667 0.33333333 0.00000000 0.00000000 0.00000000
 [2,] 0.0000000 0.07692308 0.84615385 0.00000000 0.07692308
 [3,] 0.0000000 0.00000000 0.06666667 0.46666667 0.46666667
 [4,] 0.9166667 0.08333333 0.00000000 0.00000000 0.00000000
 [5,] 0.8461538 0.15384615 0.00000000 0.00000000 0.00000000
 [6,] 0.0000000 0.00000000 0.00000000 0.00000000 1.00000000
 [7,] 0.0000000 0.15000000 0.05000000 0.10000000 0.70000000
 [8,] 0.4000000 0.50000000 0.00000000 0.00000000 0.10000000
 [9,] 0.0000000 0.05263158 0.00000000 0.05263158 0.89473684
[10,] 0.2400000 0.00000000 0.00000000 0.08000000 0.68000000
> colMeans(topicproportion4)
[1] 0.1812036 0.2487882 0.1719936 0.1225514 0.2754633
```

Appendix: correlation between topics and sentiment score for real users

```
> cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$felling)

        Pearson's product-moment correlation

data:  ENGREVIEW.T$sentiment and ENGREVIEW.T$felling
t = 26.868, df = 23527, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1601134 0.1849080
sample estimates:
     cor
0.172538

> cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$box)

        Pearson's product-moment correlation

data:  ENGREVIEW.T$sentiment and ENGREVIEW.T$box
t = -1.2064, df = 23527, p-value = 0.2277
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.020640514  0.004913172
sample estimates:
         cor
-0.007864955
```

```
> cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$action)

        Pearson's product-moment correlation

data:  ENGREVIEW.T$sentiment and ENGREVIEW.T$action
t = 20.826, df = 23527, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1219765 0.1470692
sample estimates:
      cor
0.1345444

> cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$actor)

        Pearson's product-moment correlation

data:  ENGREVIEW.T$sentiment and ENGREVIEW.T$actor
t = -36.565, df = 23527, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2439452 -0.2197639
sample estimates:
       cor
-0.2318904

> cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$series)

        Pearson's product-moment correlation

data:  ENGREVIEW.T$sentiment and ENGREVIEW.T$series
t = -12.061, df = 23527, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.09107927 -0.06568103
sample estimates:
        cor
-0.07839287
```

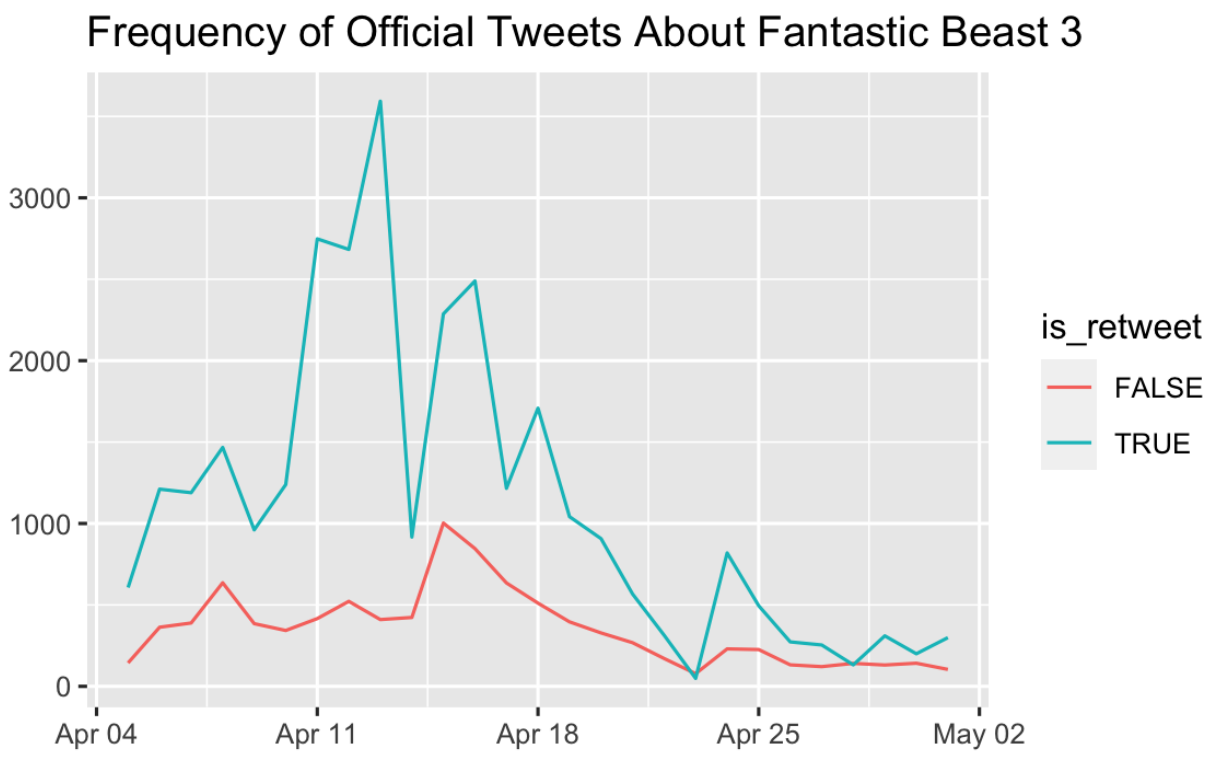
Appendix: correlation between topics and sentiment score for real users

```
> cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$performance)

        Pearson's product-moment correlation

data:  ENGREVIEW.T4$sentiment and ENGREVIEW.T4$performance
t = 6.3657, df = 13837, p-value = 2.005e-10
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.03740957 0.07063442
sample estimates:
       cor
0.05403695

> cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$general)

        Pearson's product-moment correlation

data:  ENGREVIEW.T4$sentiment and ENGREVIEW.T4$general
t = 0.35014, df = 13837, p-value = 0.7262
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.01368517  0.01963666
sample estimates:
       cor
0.00297657

> cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$release)

        Pearson's product-moment correlation

data:  ENGREVIEW.T4$sentiment and ENGREVIEW.T4$release
t = -1.7132, df = 13837, p-value = 0.0867
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.031216286  0.002098768
sample estimates:
       cor
-0.0145628
```

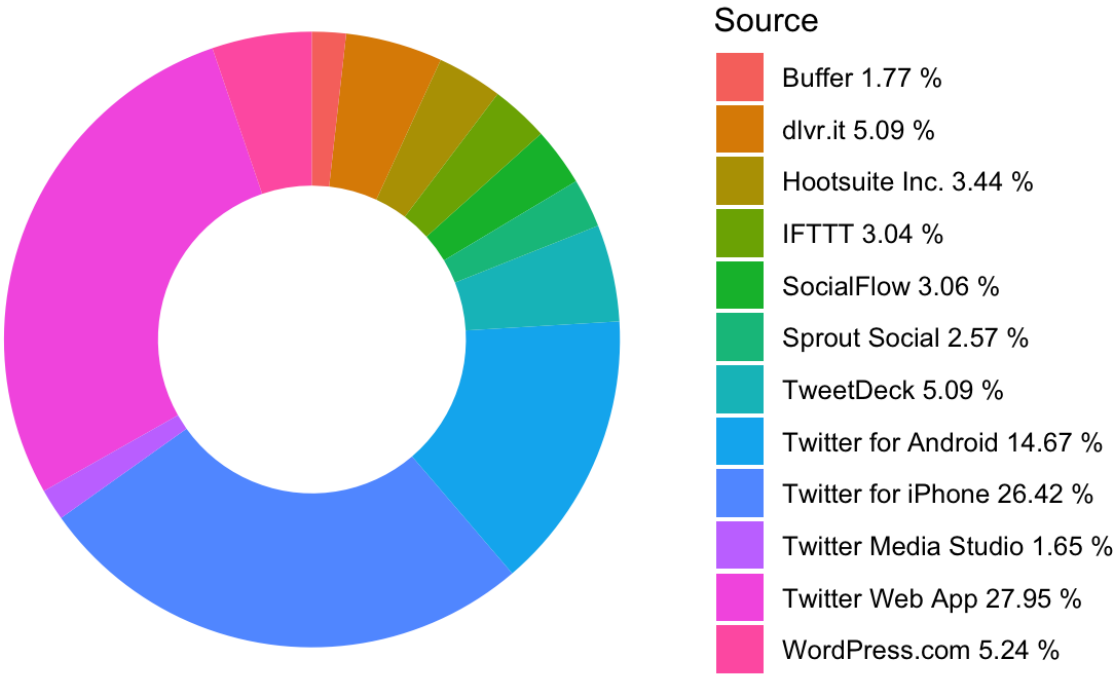
```
> cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$censor)

        Pearson's product-moment correlation

data:  ENGREVIEW.T4$sentiment and ENGREVIEW.T4$censor
t = -4.1794, df = 13837, p-value = 2.94e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.05213788 -0.01885776
sample estimates:
        cor
-0.03550766
```

```
> cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$series)

        Pearson's product-moment correlation

data:  ENGREVIEW.T4$sentiment and ENGREVIEW.T4$series
t = -2.5835, df = 13837, p-value = 0.009791
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.038604256 -0.005298197
sample estimates:
        cor
-0.02195732
```

Appendix 5: Non-user Analysis



Source: Data collected from Twitter's REST API via rtweet



Frequency of Hashtag for Non-User Accounts

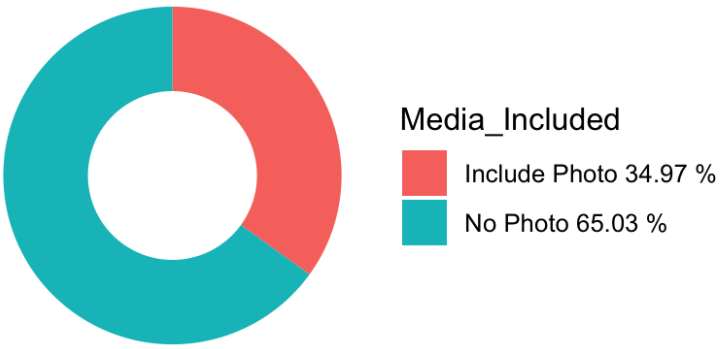
SecretsOfDumbledore	376
FantasticBeasts	188
win	117
FantasticBeasts SecretsOfDumbledore	86
FantasticBeasts SecretsOfDumbledore competition wi…	80
FantasticBeastsTheSecretsOfDumbledore	67
WarnerBrosIndia FantasticBeasts SecretsOfDumbledore	49
trakt	46
FantasticBeasts TheSecretsOfDumbledore	33
IMDb	33
SecretsOfDumbledore BoxOffice	28
SecretsofDumbledore	27
FantasticBeasts FantasticBeastsTheSecretsOfDumbled…	22
NowWatching	22
FantasticBeasts TheSecretsofDumbledore	21
SecretsOfDumbledore FantasticBeasts	20

Average Likes and Retweets for Top 10 Liked Hashtags

hashtags	favorite_count	retweet_count
IStandWithJohnnyDepp JusticeForJohnnyDeep	4253.0000	878.00000
Harrypotter Jkrowling	3068.0000	267.00000
MadsMikkelsen FantasticBeasts SecretsOfDumbledore	812.0000	201.00000
MamoruMiyano ToshiyukiMorikawa KazuhikoInoue	535.0000	125.00000
CursedChildNYC SecretsOfDumbledore	470.0000	50.00000
Beast KGFChapter2 Beast	375.0000	145.00000
SecretsOfDumbledore AtomSweeps	303.0000	90.00000
themagicofminalima FantasticBeasts SecretsOfDumbl…	290.0000	39.00000
FilmedAtWBSL SecretsOfDumbledore	194.0000	20.00000
TheSecretsOfDumbledore	177.0000	26.16667

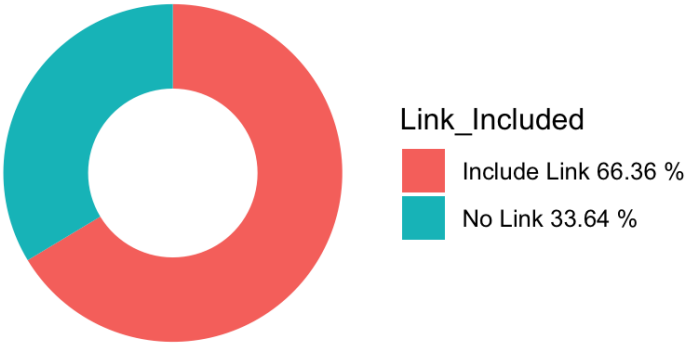
```
Call:
lm(formula = favorite_count ~ is_hashtag, data = hashtag_likes)

Residuals:
    Min      1Q  Median      3Q     Max
  -33.0   -30.0   -16.1   -15.1 11209.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   16.118      2.850   5.656  1.6e-08 ***
is_hashtag    16.837      4.707   3.577 0.000349 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 209 on 8492 degrees of freedom
Multiple R-squared:  0.001505,  Adjusted R-squared:  0.001387
F-statistic:  12.8 on 1 and 8492 DF,  p-value: 0.0003493
```

```
Call:
lm(formula = retweet_count ~ is_hashtag, data = hashtag_likes)

Residuals:
    Min      1Q  Median      3Q     Max
  -5.95   -4.95   -2.43   -2.43 2539.57

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4339     0.5768   4.220 2.47e-05 ***
is_hashtag    3.5188     0.9528   3.693 0.000223 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.31 on 8492 degrees of freedom
Multiple R-squared:  0.001604,  Adjusted R-squared:  0.001486
F-statistic: 13.64 on 1 and 8492 DF,  p-value: 0.0002227
```



```
Call:
lm(formula = favorite_count ~ is_media, data = butternut)

Residuals:
    Min      1Q  Median      3Q     Max
  -51.9   -48.9    -6.4    -5.4 11174.1

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.366      2.799   2.274    0.023 *
is_media      45.537      4.734   9.619   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 208.1 on 8492 degrees of freedom
Multiple R-squared:  0.01078,   Adjusted R-squared:  0.01066
F-statistic: 92.53 on 1 and 8492 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = retweet_count ~ is_media, data = butternut)

Residuals:
    Min      1Q  Median      3Q     Max
  -8.40   -7.40   -1.21   -1.21 2533.60

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.2096     0.5679   2.130   0.0332 *
is_media      7.1897     0.9603   7.487 7.77e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.21 on 8492 degrees of freedom
Multiple R-squared:  0.006557,  Adjusted R-squared:  0.00644
F-statistic: 56.05 on 1 and 8492 DF,  p-value: 7.766e-14
```




```
Call:
lm(formula = retweet_count ~ is_link, data = butternut)

Residuals:
    Min      1Q  Median      3Q     Max
  -4.21   -4.21   -3.48   -3.48 2537.79

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.2114     0.7922   5.316 1.09e-07 ***
is_link      -0.7351     0.9724  -0.756     0.45
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.34 on 8492 degrees of freedom
Multiple R-squared:  6.729e-05, Adjusted R-squared:  -5.046e-05
F-statistic: 0.5714 on 1 and 8492 DF,  p-value: 0.4497
```

```
Call:
lm(formula = favorite_count ~ is_link, data = butternut)

Residuals:
    Min      1Q  Median      3Q     Max
  -29.0   -26.0   -18.9   -17.9 11197.0

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   29.044      3.913   7.423 1.25e-13 ***
is_link      -10.179      4.803  -2.119   0.0341 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 209.1 on 8492 degrees of freedom
Multiple R-squared:  0.0005286, Adjusted R-squared:  0.0004109
F-statistic: 4.491 on 1 and 8492 DF,  p-value: 0.03409
```

R CODE

```
#NON-USER ANALYSIS
library(rtweet)
library(ggplot2)
library(dplyr)
library(tidyverse)

pre_release = read_twitter_csv("full name0413.csv")
pre_release2 = read_twitter_csv("full name0422.csv")
post_release = read_twitter_csv("full name0501.csv")


beast = rbind (pre_release,pre_release2,post_release)
summary(beast)
beast%>%
  dplyr::group_by(is_retweet)%>%
  ts_plot("days") +
  ggplot2::labs(
    x = NULL, y = NULL,
    title = "Frequency of Official Tweets About Fantastic Beast 3",
```

```
    caption = "\nSource: Data collected from Twitter's REST API via rtweet"
  )

#Exploratory Findings
# Remove retweets
beast_organic <- beast[beast$is_retweet==FALSE, ]
# Remove replies
beast_organic <- subset(beast_organic, is.na(beast_organic$reply_to_status_id))

#Find the ratio of replies/retweet/organic tweet
beast_retweets <- beast[beast$is_retweet==TRUE,]
# Keeping only the replies
beast_replies <- subset(beast, !is.na(beast$reply_to_status_id))

data$fraction = data$count / sum(data$count)
data$percentage = data$count / sum(data$count) * 100
data$ymax = cumsum(data$fraction)
data$ymin = c(0, head(data$ymax, n=-1))
# Rounding the data to two decimal points
data$percentage <- round(data$percentage, 2)
# Specify what the legend should say
Type_of_Tweet <- paste(data$category, data$percentage, "%")
ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Type_of_Tweet)) +
  geom_rect() +
  coord_polar(theta="y") +
  xlim(c(2, 4)) +
  theme_void() +
  theme(legend.position = "right")

#From where are the tweets published?
beast_source <- beast_organic %>%
  select(source) %>%
  group_by(source) %>%
  summarize(count=n())
beast_source <- subset(beast_source, count > 100)
data <- data.frame(
  category=beast_source$source,
  count=beast_source$count
)
data$fraction = data$count / sum(data$count)
```

```
data$percentage = data$count / sum(data$count) * 100
data$ymax = cumsum(data$fraction)
data$ymin = c(0, head(data$ymax, n=-1))
data$percentage <- round(data$percentage, 2)
#Sort by highest percentage
data <- data %>% arrange(-percentage)

Source <- paste(data$category, data$percentage, "%")
ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Source)) +
  geom_rect() +
  coord_polar(theta="y") + # Try to remove that to understand how the chart is built initially
  xlim(c(2, 4)) +
  theme_void() +
  theme(legend.position = "right")

#Which tweets receive the most engagement?
#Number of likes and number of retweets
beast_organic <- beast_organic %>% arrange(-favorite_count)
head(beast_organic$text)

beast_organic <- beast_organic %>% arrange(-retweet_count)
head(beast_organic$screen_name,10)

#Which hashtags are usually used?
beast_hashtag <- beast_organic %>%
  select(hashtags) %>%
  group_by(hashtags) %>%
  summarize(count=n())
beast_hashtag <- beast_hashtag %>% arrange(-count)

#Does using hashtags increases like?
#Create subset of just like counts and hastags
hashtag_likes <- subset(beast_organic, select =
c("text","screen_name","favorite_count","retweet_count","hashtags","verified"))
hashtag_likes$is_hashtag = NA
hashtag_likes$is_hashtag[is.na(hashtag_likes$hashtags)]<- 0
hashtag_likes$is_hashtag[!is.na(hashtag_likes$hashtags)]<- 1

reg1<-lm(favorite_count~is_hashtag, data=hashtag_likes)
summary(reg1)
```

```
reg2<-lm(retweet_count~is_hashtag, data=hashtag_likes)
summary(reg2)

hashtag_likes$hashtags = as.factor(hashtag_likes$hashtags)
summary(hashtag_likes$hashtags)
#Which hashtag generate the most engagement
nut = hashtag_likes %>% group_by(hashtags) %>%
summarise_at(vars(-text,-screen_name,-is_hashtag,-verified), funs(mean(., na.rm=TRUE)))
nut<- nut %>% arrange(-favorite_count)
nut <- subset(nut, favorite_count>100)

#With the link shared, what link source receive highest engagement
butternut <- subset(beast_organic, select =
c("text","screen_name","favorite_count","retweet_count","urls_url","media_type","location"))
butternut$is_media = NA
butternut$is_media[is.na(butternut$media_type)]<- 0
butternut$is_media[!is.na(butternut$media_type)]<- 1
summary(as.factor(butternut$is_media))
#Visualization
data <- data.frame(
  category=c("Include Link", "No Link"),
  count=c(5637, 2857)
)
# Adding columns
data$fraction = data$count / sum(data$count)
data$percentage = data$count / sum(data$count) * 100
data$ymax = cumsum(data$fraction)
data$ymin = c(0, head(data$ymax, n=-1))
data$percentage <- round(data$percentage, 2)
# Specify what the legend should say
Link_Included <- paste(data$category, data$percentage, "%")
ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Link_Included)) +
  geom_rect() +
  coord_polar(theta="y") +
  xlim(c(2, 4)) +
  theme_void() +
  theme(legend.position = "right")

butternut$is_link = NA
butternut$is_link[is.na(butternut$urls_url)]<- 0
```

```
butternut$is_link[!is.na(butternut$urls_url)]<- 1
summary(as.factor(butternut$is_link))

#Does include photo increase engagement?
reg3<-lm(favorite_count~is_media, data=butternut)
summary(reg3)
reg4<-lm(retweet_count~is_media, data=butternut)
summary(reg4)

#Does include link increase engagement?
reg5<-lm(favorite_count~is_link, data=butternut)
summary(reg5)
reg6<-lm(retweet_count~is_link, data=butternut)
summary(reg6)
```

Part 2 Code for Sentiment Analysis and Word Cloud

```
#study the tweets with external links
tweet_df$if_external_link <- ifelse(grepl("https://t.co/", tweet_df$text), 1, 0)
external_link_0 = tweet_df[grep(0, tweet_df$if_external_link),]
external_link_1 = tweet_df[grep(1, tweet_df$if_external_link),]
summary(external_link_1$favorite_count)

temp=external_link_1[order(-external_link_1$favorite_count),]
temp$text[1:5]



####Sentiment Analysis for the whole data set####
library(sentimentr)
library(textcat)
library(data.table)

language=textcat(tweet_df$text)
sort(table(language),decreasing=T)
tweet_df_eng=tweet_df[tweet_df$lang=="en",]
tweet_df_eng <- na.omit(tweet_df_eng$text)

tweet_df_text=get_sentences(tweet_df_eng)
tweet_df_sentence = sentiment(tweet_df_text)
tweet_df_sentence = as.data.table(tweet_df_sentence)
tweet_df_review=tweet_df_sentence[,
```

```
                list(
                  review.sentiment = mean(sentiment)
                ),
                by=list(element_id)]
tweet_df_sentiment = cbind(tweet_df_eng, tweet_df_review)
summary(tweet_df_sentiment$review.sentiment)
hist(tweet_df_sentiment$review.sentiment)

####Sentiment Analysis for the data set containing external links####
#with external link
language=textcat(external_link_1$text)
external_1_eng=external_link_1[external_link_1$lang=="en",]
external_1_eng <- na.omit(external_1_eng$text)

external_1_text=get_sentences(external_1_eng)
external_1_sentence = sentiment(external_1_text)
external_1_sentence = as.data.table(external_1_sentence)
external_1_review=external_1_sentence[,
                            list(
                              review.sentiment = mean(sentiment)
                            ),
                            by=list(element_id)]
external_link_1=external_link_1[external_link_1$lang=="en",]
external_1_sentiment = cbind(external_link_1, external_1_review)
summary(external_1_sentiment$review.sentiment)
hist(external_1_sentiment$review.sentiment)

#without external link
external_0_eng=external_link_0[external_link_0$lang=="en",]
external_0_eng <- na.omit(external_0_eng$text)

external_0_text=get_sentences(external_0_eng)
external_0_sentence = sentiment(external_0_text)
external_0_sentence = as.data.table(external_0_sentence)
external_0_review=external_0_sentence[,
                              list(
                                review.sentiment = mean(sentiment)
                              ),
                              by=list(element_id)]
external_link_0=external_link_0[external_link_0$lang=="en",]
```

```
external_0_sentiment = cbind(external_link_0, external_0_review)

summary(external_0_sentiment$review.sentiment)
hist(external_0_sentiment$review.sentiment)

###plot the daily average sentiment scores against each other
library(ggplot2)

# define bin (group by date)
external_1_sentiment$datetime = anytime(external_1_sentiment$created_at)
external_1_sentiment$dateGroup = format(external_1_sentiment$datetime,'%m-%d')
external_1_sentiment[is.na(external_1_sentiment$dateGroup),]

external_0_sentiment$datetime = anytime(external_0_sentiment$created_at)
external_0_sentiment$dateGroup = format(external_0_sentiment$datetime,'%m-%d')
external_0_sentiment[is.na(external_0_sentiment$dateGroup),]

external_1_sentiment <- external_1_sentiment %>% group_by(dateGroup) %>%
summarize(link="with link",mean=mean(review.sentiment),n=n())
external_0_sentiment <- external_0_sentiment %>% group_by(dateGroup) %>%
summarize(link="without link",mean=mean(review.sentiment),n=n())
total_sentiment <- rbind(external_1_sentiment, external_0_sentiment)

#plot
ggplot(total_sentiment,
       aes(x = dateGroup, y = mean, fill = link)) +
  geom_col(width = 0.8, position = 'dodge') +
  labs(title = 'Link vs. No Link Daily Average Sentiment Score',
       x = 'Group Date',
       y = 'Average Sentiment Score') +
  scale_fill_manual(values = c('#999999','#E69F00')) # customized color theme

ggplot(total_sentiment, aes(x=dateGroup, y=total_sentiment[,3], fill=link)) +
geom_bar(stat="identity", position="dodge")


####Topic Model for External Links yes/no#####
library(tm)
library(textstem)
```

```
library(RColorBrewer)
library(wordcloud)

library(lda)
library(topicmodels)

###word cloud for texts with external links
#add self-defined stopwords
myStopwords = c(stopwords("english"),"fantastic","beasts","beast",
"fantasticbeasts","secret","dumbledore","secretofdumbledore") #add stop words

#preprocessing data
doc_link_1 = VCorpus(VectorSource(tweet_df$text[tweet_df$if_external_link==1]))
doc_link_1 <-tm_map(doc_link_1,content_transformer(tolower))
doc_link_1 = tm_map(doc_link_1, removePunctuation)
doc_link_1 = tm_map(doc_link_1, removeNumbers)
doc_link_1 = tm_map(doc_link_1, removeWords, myStopwords)
doc_link_1 = tm_map(doc_link_1, removeWords, stopwords("english"))
doc_link_1 = tm_map(doc_link_1, stripWhitespace)
doc_link_1 = tm_map(doc_link_1, content_transformer(lemmatize_strings))

DTM_link_1 = DocumentTermMatrix(doc_link_1)

#wordcloud generation
#frequency
matrix_DTM_link_1 =as.matrix(DTM_link_1)
WD_link_1 = sort(colSums(matrix_DTM_link_1),decreasing=TRUE)

#transform the data to fit with the wordcloud package
wcloud_link_1=data.table(words=names(WD_link_1),freq=WD_link_1)

set.seed(1)
#adjust focus sizes with scale
par(mar=c(1,1,1,1))
wordcloud(words = wcloud_link_1$words, freq = wcloud_link_1$freq,
        scale=c(5,1),
        min.freq = 2,
        max.words=200,
        random.order=FALSE,
        rot.per=0.1,
```

```
        colors=brewer.pal(8, "Dark2"))


###word cloud for texts without external links
#preprocessing data
doc_link_0 = VCorpus(VectorSource(tweet_df$text[tweet_df$if_external_link==0]))
doc_link_0 <-tm_map(doc_link_0,content_transformer(tolower))
doc_link_0 = tm_map(doc_link_0, removePunctuation)
doc_link_0 = tm_map(doc_link_0, removeNumbers)
doc_link_0 = tm_map(doc_link_0, removeWords, myStopwords)
doc_link_0 = tm_map(doc_link_0, removeWords, stopwords("english"))
doc_link_0 = tm_map(doc_link_0, stripWhitespace)
doc_link_0 = tm_map(doc_link_0, content_transformer(lemmatize_strings))
DTM_link_0 = DocumentTermMatrix(doc_link_0)

#wordcloud generation

#frequency
matrix_DTM_link_0 =as.matrix(DTM_link_0)
WD_link_0 = sort(colSums(matrix_DTM_link_0),decreasing=TRUE)

#transform the data to fit with the wordcloud package
wcloud_link_0=data.table(words=names(WD_link_0),freq=WD_link_0)

set.seed(1)
#adjust focus sizes with scale
par(mar=c(1,1,1,1))
wordcloud(words = wcloud_link_0$words, freq = wcloud_link_0$freq,
        scale=c(5,1),
        min.freq = 2,
        max.words=200,
        random.order=FALSE,
        rot.per=0.1,
        colors=brewer.pal(8, "Dark2"))
```

Part 3 Code for bass, topic model and correlation

```
library(rtweet)
library(ggplot2)
library(dplyr)
library(diffusion)
```

```
library(sentimentr)
library(textcat)
library(data.table)
library(tm)
library(textstem)
library(RColorBrewer)
library(wordcloud)
library(lda)
library(topicmodels)
library(doBy)
library(anytime)
library(textclean)



####for non users#######
full_0413 = read_twitter_csv("full name0413.csv")
full_0422 = read_twitter_csv("full name0422.csv")
full_0501 = read_twitter_csv("full name0501.csv")
ful1=rbind(full_0413,full_0422)
FB3_full=rbind(ful1,full_0501)
length(unique(FB3_full$user_id))

#####for real users
load("D:/1.JHU/4. spring 2/social media/group/beastkeke2.RData")
FB3=as.data.frame(beast_keke2)
names(FB3)
#check the sources of the original dataset
table(FB3$source)
FB3 %>%
  group_by(FB3$source) %>%
  summarize(Count=n()) %>%
  mutate(Percent = round((Count/sum(Count)*100))) %>%
  arrange(desc(Count))

#clean by sources and keep those tweets only from Twitter for purpose of this study
keep <- c("Twitter for iPhone", "Twitter for Android", "Twitter Web App", "Twitter for iPad")
FB3_full25 = subset(FB3, source %in% keep)

length(unique(FB3$user_id))
```

```
####basic assumption
temp.s = FB3_full25
external_link = temp.s[grep("https://t.co/",temp.s$text), ]
summary(external_link$favorite_count)
FB3_full25$if_external_link <- ifelse(grepl("https://t.co/", FB3_full25$text), 1, 0)
noexternal_link = FB3_full25[grep("0",FB3_full25$if_external_link),]


###########Bass Model##########
FB3_full$datetime = anytime(FB3_full$create)
FB3_full$day = format(FB3_full$datetime,"%m%d")

ts_plot(FB3_full,by = "days")
bass=as.data.frame(FB3_full %>% count(day))

plot(bass)

newdata=bass$n[-1]
newdata=as.data.frame(newdata)
newdata$ts=newdata$newdata
fit = diffusion(newdata$ts,type="bass")
fit
par(mar=c(3,3,3,3))
plot(fit,cumulative=FALSE)
lines(newdata$ts)




####bass model for real users#####
external_link$datetime = anytime(external_link$created_at)
external_link$day = format(external_link$datetime,"%m%d")

ts_plot(external_link,by = "day")
bass=as.data.frame(external_link %>% count(day))

plot(bass)

newdata2=bass$n[-1]
newdata2=as.data.frame(newdata2)
```

```
newdata2$ts=newdata2$newdata
fit = diffusion(newdata2$ts,type="bass")
fit
par(mar=c(3,3,3,3))
plot(fit,cumulative=FALSE)
lines(newdata2$ts)

######sentiment analysis#######
#####sentiment with external link###
language=textcat(external_link$text)
sort(table(language),decreasing=T)
ENGREVIEW=external_link[external_link$lang=="en",]

MYTEXT=get_sentences(ENGREVIEW$text)
SENTENCE.S = sentiment(MYTEXT)
SENTENCE.S = as.data.table(SENTENCE.S)
REVIEW.S=SENTENCE.S[,
                    list(
                      review.sentiment = mean(sentiment)
                    ),
                    by=list(element_id)]
ENGREVIEW.S = cbind(ENGREVIEW, REVIEW.S)
summary(ENGREVIEW.S$review.sentiment)
hist(ENGREVIEW.S$review.sentiment)


ENGREVIEW.S$datetime = anytime(ENGREVIEW.S$created_at)
ENGREVIEW.S$day = format(ENGREVIEW.S$datetime,'%d')

temp2 = as.data.table(ENGREVIEW.S)
DAY.S = temp2[,list(
                      meansentiment=mean(review.sentiment)
                    ),
                    by=day]



DAY.S = DAY.S[order(DAY.S$day),]
par(mar=c(3,3,3,3))
barplot(DAY.S$meansentiment, names.arg=DAY.S$day)
```

```
####sentiment w/o external link#####

language2=textcat(noexternal_link$text)
sort(table(language),decreasing=T)
ENGREVIEW2=noexternal_link[noexternal_link$lang=="en",]

MYTEXT2=get_sentences(ENGREVIEW2$text)
SENTENCE.S2 = sentiment(MYTEXT2)
SENTENCE.S2 = as.data.table(SENTENCE.S2)
REVIEW.S2=SENTENCE.S2[,
                list(
                  review.sentiment = mean(sentiment)
                ),
                by=list(element_id)]
ENGREVIEW.S2 = cbind(ENGREVIEW2, REVIEW.S2)
summary(ENGREVIEW.S2$review.sentiment)
hist(ENGREVIEW.S2$review.sentiment)


ENGREVIEW.S2$datetime = anytime(ENGREVIEW.S2$created_at)
ENGREVIEW.S2$day = format(ENGREVIEW.S2$datetime,'%d')

temp22 = as.data.table(ENGREVIEW.S2)
DAY.S2 = temp22[,list(
  meansentiment=mean(review.sentiment)
),
by=day]



DAY.S2 = DAY.S2[order(DAY.S2$day),]
par(mar=c(3,3,3,3))
barplot(DAY.S2$meansentiment, names.arg=DAY.S2$day)



########TOPIC analysis#######

temp3=ENGREVIEW2$text
```

```
temp3=replace_html(temp3)
temp3=gsub("^RT", "", temp3)
temp3=tolower(temp3)
temp3=replace_html(temp3)
temp3=replace_url(temp3)
temp3=replace_hash(temp3)
temp3=replace_emoji(temp3)
temp3=replace_emoticon(temp3)
temp3=replace_tag(temp3)
temp3=replace_non_ascii(temp3)
temp3=replace_symbol(temp3)

docs = VCorpus(VectorSource(temp3))

# Preprocess the data:
docs <-tm_map(docs,content_transformer(tolower))
docs = tm_map(docs, removePunctuation)
docs = tm_map(docs, removeNumbers)
docs = tm_map(docs, removeWords, stopwords("english"))
docs = tm_map(docs, stripWhitespace)

# lemmatize the words
docs = tm_map(docs, content_transformer(lemmatize_strings))

#convert data into document term matrix format
DTM = DocumentTermMatrix(docs)

# Generate wordcloud for all the documents
# Calculate the frequency of each word
DTMMATRIX=as.matrix(DTM)



###### topic models to identify common themes #######
# Transform the data to fit with the topic model package
input = dtm2ldaformat(DTM)

# set the random seed so results are replicable
set.seed(12345)
K=5
N=1000
```

```
result = lda.collapsed.gibbs.sampler(
  input$documents,
  K,              # The number of topics.
  input$vocab,
  N, # The number of iteration
  alpha=1/K,         # The Dirichlet hyper parameter for topic proportion
  eta=0.1,           # The Dirichlet hyper parameter for topic multinomial
  compute.log.likelihood=TRUE)

TOPIC = top.topic.words(result$topics, 10, by.score=TRUE)
TOPIC

plot(result$log.likelihoods[1,],type="o")

# Count number of topic keywords for each sentence
T1=t(result$document_sums)
# Calculate the topic assignment for each sentence
topicproportion=T1/rowSums(T1)
topicproportion[1:10,]
colMeans(topicproportion)



#combine the topic proportion data with the original data
# Combine the sentiment score with the original data
ENGREVIEW.T = cbind(ENGREVIEW2, topicproportion)



names(ENGREVIEW.T)[names(ENGREVIEW.T) == "1"] <- "felling"
names(ENGREVIEW.T)[names(ENGREVIEW.T) == "2"] <- "box"
names(ENGREVIEW.T)[names(ENGREVIEW.T) == "3"] <- "action"
names(ENGREVIEW.T)[names(ENGREVIEW.T) == "4"] <- "actor"
names(ENGREVIEW.T)[names(ENGREVIEW.T) == "5"] <- "series"

ENGREVIEW.T$sentiment = ENGREVIEW.S2$review.sentiment

# Evaluate the relationship with between topic proportion and sentiment score

cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$felling)
cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$box)
cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$action)
```

```
cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$actor)
cor.test(ENGREVIEW.T$sentiment,ENGREVIEW.T$series)

#####topic For promoters ########

temp4=ENGREVIEW$text
temp4=replace_html(temp4)
temp4=gsub("^RT", "", temp4)
temp4=tolower(temp4)
temp4=replace_html(temp4)
temp4=replace_url(temp4)
temp4=replace_hash(temp4)
temp4=replace_emoji(temp4)
temp4=replace_emoticon(temp4)
temp4=replace_tag(temp4)
temp4=replace_non_ascii(temp4)
temp4=replace_symbol(temp4)

docs4 = VCorpus(VectorSource(temp4))

# Preprocess the data:
docs4 <-tm_map(docs4,content_transformer(tolower))
docs4 = tm_map(docs4, removePunctuation)
docs4 = tm_map(docs4, removeNumbers)
docs4 = tm_map(docs4, removeWords, stopwords("english"))
docs4 = tm_map(docs4, stripWhitespace)

# lemmatize the words
docs4 = tm_map(docs4, content_transformer(lemmatize_strings))

#convert data into document term matrix format
DTM4 = DocumentTermMatrix(docs4)

###### topic models to identify common themes #######
# Transform the data to fit with the topic model package
input4 = dtm2ldaformat(DTM4)

# set the random seed so results are replicable
set.seed(12345)
K=5
```

```
N=1000
result4 = lda.collapsed.gibbs.sampler(
  input4$documents,
  K,              # The number of topics.
  input4$vocab,
  N, # The number of iteration
  alpha=1/K,         # The Dirichlet hyper parameter for topic proportion
  eta=0.1,           # The Dirichlet hyper parameter for topic multinomial
  compute.log.likelihood=TRUE)

TOPIC4 = top.topic.words(result4$topics, 10, by.score=TRUE)
TOPIC4

plot(result4$log.likelihoods[1,],type="o")

# Count number of topic keywords for each sentence
T14=t(result4$document_sums)
# Calculate the topic assignment for each sentence
topicproportion4=T14/rowSums(T14)
topicproportion4[1:10,]
colMeans(topicproportion4)



#combine the topic proportion data with the original data
# Combine the sentiment score with the original data
ENGREVIEW.T4 = cbind(ENGREVIEW[-c(1:9),], topicproportion4)

names(ENGREVIEW.T4)[names(ENGREVIEW.T4) == "1"] <- "general"
names(ENGREVIEW.T4)[names(ENGREVIEW.T4) == "2"] <- "performance"
names(ENGREVIEW.T4)[names(ENGREVIEW.T4) == "3"] <- "release"
names(ENGREVIEW.T4)[names(ENGREVIEW.T4) == "4"] <- "censor"
names(ENGREVIEW.T4)[names(ENGREVIEW.T4) == "5"] <- "series"

ENGREVIEW.S=ENGREVIEW.S[-c(1:9),]
ENGREVIEW.T4$sentiment = ENGREVIEW.S$review.sentiment

# Evaluate the relationship with between topic proportion and sentiment score

cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$general)
cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$performance)
```

```
cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$release)
cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$censor)
cor.test(ENGREVIEW.T4$sentiment,ENGREVIEW.T4$series)
```