Predicting Upvotes and Replies on New York Times Article Comments

Natthawut Boonsiriphatthanajaroen
Princeton University
natthab@princeton.edu

Jean Luo
Princeton University
jeanluo@princeton.edu

Maxwell Xu
Princeton University
mmxu@princeton.edu

Abstract

The New York Times is one of the major news outlets in the U.S. and undoubtedly plays a role in shaping public opinion on current events. The comment section gives information about many readers' opinions, and the number of replies and upvotes on these comments is a reflection of how people respond to certain opinions or what kinds of sentiments are most popular or controversial among New York Times readers. The dataset we use includes information on 231450 comments posted in January 2017 on a variety of articles. The data provides the text of the comments themselves as well as 34 features, 2 of which we are interested in analyzing: number of upvotes and number of replies for each comment. We reduce a bigram bag of words for our dataset of comments to 18 LDA topics and use these topics as well as supervised learning methods including linear regression, ridge regression, random forest regression, Naive Bayes, and logistic regression, to predict number of upvotes and replies.

1 Introduction

The New York Times is a newspaper that has worldwide influence and readership and is ranked 2nd most circulated in the U.S. The comments section of articles is quite active and gives a glimpse into reader's opinions about the subject matter and ideas presented in articles. In this work, we are interested to uncover patterns in the words that readers use in comments and explore how effective these are in predicting the number of upvotes and replies the comment gets, which are indications of how well-received they are. The comments dataset has a feature called *recommendations* which stores the number of upvotes a comments received, and another called *replyCount* that stores the number of replies it got. [1]

2 Related Work

Currently, the New York Times has human moderators that have to sift through around 12,000 comments per day. There are on average 700 comments per article. With such a massive amount of comments the newspaper must deal with, machine learning methods are useful in categorizing and filtering comments to reduce the workload for human moderators. According to Hosseini et al., the New York Times plans to adopt an API aimed at identifying toxic comments, defined as rude and inappropriate, as a first pass [2]. Such rude and inappropriate comments are typically also likely to garner a large number of responses and start a discussion. Thus, we think that examining the number of replies and upvotes a comment has can be valuable in identifying potentially toxic comments as well. Sakshi Gupta (2018) uses data from the same source as our dataset and attempts to predict the popularity of New York Times comments by converting the variable containing the number of upvotes into a binary variable so that a comment is considered popular if it has 4 or more upvotes and not popular otherwise.[4] In her work, she uses sentiment analysis to explore correlations between the type of news the comment references and its popularity. In our project, we use unsupervised learning methods to first uncover hidden topic structures in the comments, and based on these findings, develop and evaluate models to expand on previous work.

There are a number of challenges to topic modeling analysis. One of the problems with unsupervised learning techniques like Latent Dirichlet Allocation is that it's often unclear exactly how many latent topics exist within the corpus. Teh et al. found that using a Hierarchical Dirichlet Process as an extension of LDA can determine the optimal number of topics to use with LDA. [5]

In addition, the bag-of-words representation that discards word order and punctuation may not accurately represent the text data. Wallach found that adding larger n-gram features to a bag-of-words representation can help improve the quality of topic modeling. [6] However, Pang and Lee found that using unigrams alone often had greater accuracy in sentiment analysis, raising questions about the value of larger n-grams in text analysis. Pang and Lee also found that certain punctuation marks were useful in the sentiment analysis task, suggesting that tokenizing punctuation may be useful for topic modeling as well. [7]

3 Methods

We used unsupervised learning methods to uncover hidden topics in the text of these comments and then supervised learning methods to predict the number of upvotes and replies to comments based on the presence of words associated with latent users uncovered in the first step.

After cleaning up and preprocessing text from the comments, we ran LDA to find words associated with latent users. We compared the following classifiers:

1. Logistic Regression
2. Naive Bayes
3. Linear Regression
4. Random Forest Regressor

Step 1: Preprocessing

First, we cleaned the text data by removing HTML formatting and ignoring non-ASCII characters. Then, we edited the preprocessSentences.py script provided in Assignment 1 to tokenize the ? and ! punctuation characters and to process bigrams as well. Running this script on the comments, using only the tokens that appeared 100 or more times, converted the text data of the comments to a bag-of-words representation with 11493 features. These features included both unigrams and bigrams.

Step 2: Latent Dirichlet Allocation

The optimal number of topics was determined by a Hierarchical Dirichlet Process (using gensim's HdpModel implementation [8]) to be about 18 topics. HDP is a mixed-membership model that infers the number of topics from data. We then initialized a Latent Dirichlet Allocation model to reduce the dimensionality of the text data to 18 features. One advantage of using LDA is that it accounts for the fact that a single word can have multiple meanings and could belong to multiple topics. Other words in the document can act as a prior when inferring the most likely topic the word belongs to. One assumption LDA makes is that the ordering of words in a document does not matter. By using both unigrams and bigrams, we attempted to account for the ordering of words. We then examined the important words of each topic. The first 5 topics produces the histogram shown in Figure 1 below. We can see that none of the topics come very close to completely describing any comments, although in this graph, topic 5 appears to account for the highest proportion of the largest number of comments. In Figure 2, the y-axis is the number of comments and the x-axis is the feature pseudocount for topic 8.

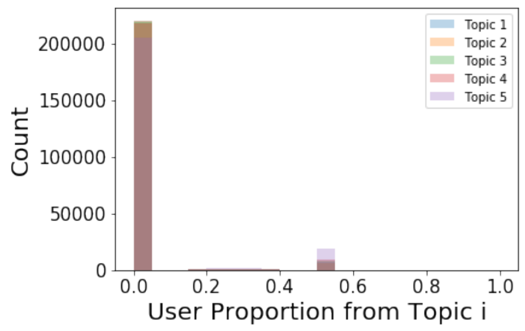

Figure 1: Pseudocounts for first 5 topics

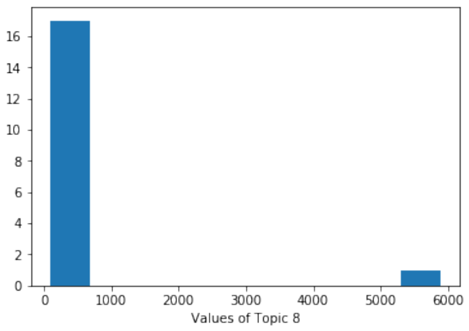

Figure 2: Histogram for Topic 8

The result of this step was a .csv file that contained a 231550-by-18 matrix that contained the pseudocounts of each topic for all the comments. We split up the comments equally into a training and testing set by assigning the even-numbered rows as x-train and the odd-numbered rows as x-test.

Step 3: Building Models

First, we examined the data and found that the number of upvotes had a range between 0 and 9279, with a mean of 22.98 and a median of 4, while the number of replies had a range between 0 and 529, with a mean of 0.538 and a median of 0. Thus, both distributions are extremely skewed to the right, and there are a large number of zeros in both the reply and upvote data. Thus, we were faced with the problem of how to divide up the distribution. We trained and fit our classifiers for both the continuous data and on the data after we split it into 4 bins and compared the two. We used the following supervised learning methods. All were run using their default parameters in *sklearn*:

1. *Regression:*
   - Linear
   - Ridge
   - Lasso
   - Random Forest
2. *Classification*
   - Naive Bayes
   - Logistic Regression

3.1: Ridge and Lasso Regression Lasso and Ridge regression are regularization methods that constrain coefficient estimates to zero. Lasso regression uses an L1 norm penalty and is considered ideal when there are many useless variables (sparse models), whereas ridge regression uses an L2 norm penalty and is more useful when all variables are useful in prediction (dense models). This is because Lasso tends to pick one correlated variable and eliminates the others, while ridge shrinks the parameters of all correlated variables together. In this case, we would expect Lasso to be more useful, but we run both to compare.

3.2: Random Forest Regressor

The background behind this model is to combine all decisions from the tree regressions. For example, this method would train M tree regressions on different sets of data, and average the estimates as shown in the equation:

$$f(X) = \sum_{m=1}^{M} \frac{1}{M} f_m(x)$$

where f_m is the m^{th} tree. The purpose of combination is to reduce the variance of an estimate. When we build our Random Forest Regression model, we import the function from the *sklearn* and fit the model with the preprocessed training data. According to this model, its variance would finally become:

$$\rho\sigma^2 + \frac{1-\rho}{M}\sigma^2$$

where M is the number of trees, σ is the variance of each tree, and ρ is the correlation between trees. From the expression, as we increase M, the latter term would disappear. The Random Forest algorithm will reduce the correlation between the trees in the tree-growing process in the random selection of input variables.

For tree regression, the model aims to branch out the options corresponding to the given depth of the tree, and splitting function follows:

$$(j^*, t^*) = arg \min_{j \in \{1,\ldots,D\}} \min_{t \in \tau_j} cost(\{\mathbf{x}_i, y_i : x_{ij} \leq t\}) + cost(\{\mathbf{x}_i, y_i : x_{ij} > t\})$$

for training samples x_i and training response y_i, where the function $cost$ is:

$$cost(\mathcal{D}) = \sum_{i \in \mathcal{D}} (y_i - \bar{y})^2$$

For the random forest regressor predicting upvotes, the top five most important features were determined. The top most important feature is described below.

Word	Pseudocount
war	10374.601627
u	8205.039748
american	3919.761405
countri	3728.407896
militari	3439.962611
would	2957.018709
kill	2825.676229
israel	2616.040503
peopl	2578.197738
state	2561.455281
iraq	2289.783596
refuge	2175.854734
?	2153.017356
gun	2147.987170
one	2072.512581
world	2028.905430
vietnam	1899.605989
europ	1850.664096
east	1847.082336
nation	1721.525100

Three representative comments of this topic are:

”Well: I suppose if the Pals cease to recognize Israel that Jewish State will disappear–just as the world’s recognition of a Palestinian State has made that state a reality.”

”By the way, when Obama allowed the resolution re the ”settlements” to pass, he effectively negated Israel’s claims to Jerusalem.”

”Moving the US Embassy to the Holy City will just reverse that resolution.”

Below is a table of the first 7 topics. Examining the words that describe these topics, we can see a theme in comment subject material for some of these topics. For instance, topic 1 seems to be about trump, topic 2 about fake news, topic 5 about education, and topic 7 could be about the muslim ban.

Topic	Sample of Important Words
1	trump, president, new, go, time, !
2	trump, news, medium, lie, nyt, fake news
3	?, get, law, ? ?, order, congress
4	like, one, look, think, see, read, article
5	school, public, education, would, use, time
6	republican, democrat, senate, supreme court, nominee
7	trump, country, tax, return, ban, muslim
8	war, american, country, military, would, israel

The second most important topic was topic 7, for which a representative comment was, ”So why not ban immigrants from Saudi Arabia where most of the 9/11 hijackers originated?”

The top two most important topics appear to be about war and national security, suggesting that New York Times commenters are most concerned with issues of public safety and will be more likely to upvote commends on such topics.

It is important to note that even though we used both unigrams and bigrams, most of the words (or word stems) that made it into the top five most important topics were unigrams, suggesting that unigrams alone may have same or better effects. For instance, ”supreme court” appeared in one topic, but so did ”supreme” and ”court” separately later in the list. [3]

3.2: Classifier

While analyzing the number of replies, we may consider classification method instead because the number of replies vary, starting from 0 up to greater than 500. In addition, the numbers of replies are also not evenly distributed. Hence, before building models, we could categorize them in groups. For example, we may say that comments that have replies less than 5 fall into group 1, replies greater than 5 but less than 10 fall into group 2, and so on. This would prepare data for classification well and could be analyzed by Naive-Bayes, and Logistic Regression.

Step 4: Evaluation

Step 4.1: Regression Analysis

The table below shows the R^2, or accuracy, and MSE values for regression models. The ”Raw” accuracy and MSE columns are the results of using the original number of reply counts. Because the number of reply and upvote counts greatly vary, we split up the distributions and did analyses on these categories. We replicated the method of Gupta (2018) and used the single cutoff of 4 replies and did the same for upvotes to binarize the data. We compared this with a different method, for which we made 5 categories: less than 5, between 5 and 10, between 10 and 15, between 15 and 20, and above 20. The ”Range” column shows the result of analyzing the data after being categorized. Accuracies were obtained using the built-in *score* method in *sklearn*.

Model	Raw R-squared	Raw MSE	Binary R-squared	Binary MSE	Range R-squared	Range MSE
Ridge	0.001121	6.149	0.00145	0.0238	-0.0101	0.0504
Lasso	-3.910e-5	6.156	-2.284e-6	0.0238	-0.0100	0.0503
Linear	0.001120	6.153	0.00145	0.0238	-0.0101	0.0504
Random Forest	-0.3106	8.068	-0.140	0.0271	-0.2096	0.06030

Table 1: Performance of classifiers for number of replies

Model	Raw R-squared	Raw MSE	Binary R-squared	Binary MSE	Range R-squared	Range MSE
Ridge	0.003585	15202.81	0.0134	0.246	0.00341	2.304
Lasso	-6.154e-7	15257.513	-4.676e-7	0.249	-0.00787	2.330
Linear	0.00358	15202.806	0.0134	0.246	-0.0316	2.304
Random Forest	-0.1527	17587.105	-0.106	0.276	-0.0985	2.539

Table 2: Performance of classifiers for number of upvotes

As shown above, though the regression methods improved the mean square errors significantly, in many cases the accuracy became worse. For predicting both upvotes and replies, using the binary data yielded better measures of accuracy and mean-squared error compared to using continuous and categorized data. Overall, Lasso and ridge regression models performed similarly, though Lasso tended to yield slightly worse accuracy measures. This is expected, however, because R^2 will always increase with more features added to the model. After exploring the data further, we were interested to see if there would be improvements in the performance of our models if we dropped insignificant comments, or, in other words, comments that users are not interested in and that have a low number of replies.

The table below shows the R^2 value of the regression model after setting a threshold and modeling for cross-validation of ratio 0.2. For example, the threshold = 0 means that we ignore the comments that have zero number of replies. The best R^2 is generated when comments with less than 2 replies are ignored. Using such a threshold could improve the model because otherwise the model would be highly influenced by the comments with zero replies, and would predict the number of replies to be close to zero, which may be unreasonable.

Model	No Threshold	Threshold = 0	Threshold = 1	Threshold = 2
Ridge	0.00082	0.00198	0.0004	0.0034
Linear	0.00082	0.00198	0.0004	0.0035
Lasso	-0.00001	-6.73e-7	-4.69e-7	-0.00056
Random Forest	-0.133	-0.330	-0.330	-0.072

Table 3: Accuracies of classifiers for number of replies after adjustments

If we set our threshold to be low, then our prediction would be influenced by low reply count, but if we set our threshold to be higher we could get more negative R-squared. However, this also comes with more meaningful interpretation. For example, when the threshold is 2, the model predicts that the maximum number of replies would be 9.4. This number is not realistic because there are some other comments that have more than 10 replies. However, after increasing the threshold, the maximum number of replies increases, because we only consider those comments with relatively higher replies. In addition, the number of replies we predict will vary more, which is what we want it to be since it is similar to real data. For example, the threshold value with 10 would give maximum replies of 32.22. This comment falls under the topic described by ”vote”, ”election”, ”trump”, etc.

Step 4.2: Classification Method

Model	Binary (cutoff 4)	Binary (cutoff 20)	Range
Bernoulli	0.976	0.976	0.994
Multinomial	0.976	0.976	0.994
Logistic	0.976	0.976	0.994

Table 4: Accuracies of classifiers for number of replies using binary versus range data

Model	Binary (cutoff 4)	Binary (cutoff 20)	Range
Bernoulli	0.523	0.851	0.731
Multinomial	0.538	0.851	0.731
Logistic	0.545	0.851	0.731

Table 5: Accuracies of classifiers for number of upvotes using binary versus range data

We tried two different cutoffs to binarize upvote and reply count data. We found that splitting the reply count data into 5 ranges slightly improved the accuracy scores of the models compared to using binary data. Since the range in upvotes is even larger than in replies, it makes sense that using a larger cutoff may be beneficial and may represent a more precise measure of popularity. In upvote data, we see that the binary data with a cutoff of 20 upvotes produced the highest accuracy measures.

Below are the ROC curves for our classification methods using the binarized data with a cutoff of 4. Interestingly, these graphs show that Multinomial Naive Bayes and Logistic Regression performed better than the Bernoulli Naive Bayes model for predicting upvotes but not for replies.

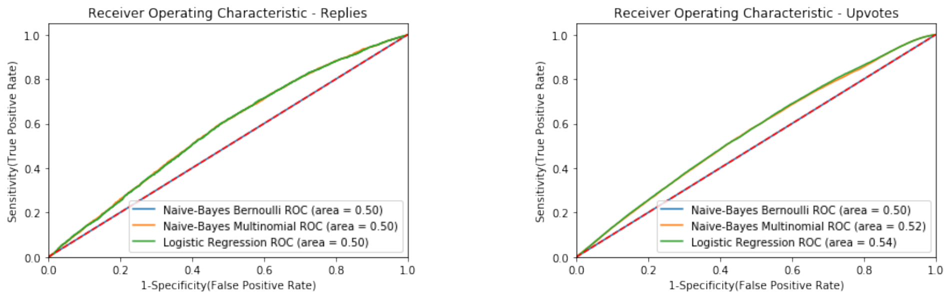

Figure 3: ROC curves for binary data (cutoff 4) for replies and upvotes

Similar to regression, we also explored the result of deleting those comments with no replies. Because comments with no replies take up 80% of all data, simply predicting the mean every time could produce a high accuracy, and building models without removing the zeros could result in a model with incredibly high accuracy but little meaning. To find an appropriate cut off, we note that the number of comments with a reply count greater than 5 is 4173 out of more than 200,000 comments. Hence, we start our analysis by ignoring comments that have less than 5 replies. Here is the result:

Model	No Threshold	Threshold = 5	Threshold = 10
Bernoulli	0.994	0.771	0.818
Multinomial	0.994	0.771	0.818
Logistic	0.994	0.771	0.818

Table 6: Accuracies of classifiers for number of replies after adjustments

Table above shows the result of classification when different thresholds are set. Note that when there is no threshold, the accuracy is incredibly high; however, when we only interested in the number of replies that are more ”meaningful”, which means they are topics that people are willing to talk about, we could see improvements.

4 Discussion and Conclusion

In this project, we fit a Latent Dirichlet Allocation model to find hidden topic structures in the words used in New York Times article comments and used these topics to predict number of upvotes and replies. While previous work had attempted to predict number of replies to a comment using features such as the length and sentiment of the comment, we introduced topic modelling prior to prediction of upvotes and replies.

We saw that splitting up the reply and upvote counts into 5 ranges improved the mean-squared error for our regression models compared to using the raw data, though the results of using the binary data performed the best. For our classification models, however, using the categorized data improved accuracy over using binary data created with a cutoff of 4 (done by previous related work), but we saw that using binary data with a cutoff of 20 for number of upvotes yielded the best performance.

If we had more time, we would have looked deeper into methods for tuning parameters of LDA, since Hierarchical Dirichlet Process may not be ideal when the number of topics is low.

It may also have been useful to look into more evaluation metrics of our supervised learning methods. Since we decided to categorize data and not restrict ourselves to dealing with binary data, many of the usual measures of classification models such as ROC-curves, precision and recall, etc. were not applicable. If we had more time, we would also have looked into using a Generalized Linear Model.

A possible extension of this project would be to explore which topics, when they account for a large proportion of a comment, correspond to a large number of upvotes and replies to assess popularity of comment themes. If compared with more conservative news outlets, this analysis could give insight into what kind of news or topics liberal-leaning readers tend to be interested in compared to more conservative readers. Looking at these comments in conjunction with the articles on which they were posted could give more insight perhaps even into whether there is bias in the way the New York Times presents its news.

References

[1] Aashita Kesarwani New York Times Comments, https://www.kaggle.com/aashita/nyt-comments, 2018.

[2] Hossein Hosseini, Sreeram Kannan, Baosen Zhang and Radha Poovendran *Deceiving Googles Perspective API Built for Detecting Toxic Comments.* University of Washington, Seattle, WA. 2017.

[3] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, 2012.

[4] Sakshi Gupta *Predicting Popularity of The New York Times Comments*. Towards Data Science, 2018.

[5] Yee Teh et al. *Hierarchical Dirichlet Process*. Berkeley, 2005.

[6] Hanna Wallach *Topic Modeling: Beyond Bag-of-Words* ICML '06 Proceedings of the 23rd international conference on Machine learning

[7] Bo Pang and Lilian Lee *Thumbs up?Sentiment Classification using Machine Learning Techniques* Proceedings of the Conference on Empirical Methods in Natural Language Processing

[8] Gensim: Topic modeling for humans *Hierarchical Dirichlet Process* https://radimrehurek.com/gensim/models/hdpmodel.html