Methods

1. Data Sets

I utilized the official API praw released by reddit.com to extract information of posts in r/worldnews community. However, praw only allows users to get 1000 observations from a single call, which means if I extract data from the top ranking of r/worldnews I can only get information of the 1000 top posts. The limited size might lead to unobserved bias in the proceeding analysis. Hence, I extracted data from a Kaggle competition instead. In the dataset, it claims to have included basic information of posts on r/worldnews released from 2008 till November 2016. However, the datasets only records news title of each post but not unique post id, so I leveraged the title of news articles to get detailed post information with praw. I only keep the title of the posts which were created in 2014 and 2015 and filtered the entries which were recorded to have 50 upvotes or more. The exact number can be seen in Table 1.

I first used the titles of news articles extracted from the Kaggle dataset to search post id of each article. Though it is possible there might be multiple or none post with the exact title of a given news article, most of the titles only have one submission result returned by praw. For some unknown algorithm confounding issue, a handful of different news titles turn out to have the same post id. I drop all the duplicate post ids and the exact number can be seen in Table 1. After obtaining the post id, I scraped post information including original article url, upvotes, author (the id of user who post the article on r/worldnews), date of created and number of comments. And I also scrape all the comments of each post, including comment body and comment author id. In order to get content information of news articles, I also used the API newsarticle3k to get all the text and author information of news from most of the given urls. After being preprocessed to match post id of both reddit and news article datasets, 19874 observations are left.

Table 1. Number of observations in the two

reddit	year	Kaggle dataset	praw dataset	article dataset	preprocessing
r/worldnews	2014	92030	10743	(many published dates missing)	9154
r/worldnews	2015	94621	11947		10720
totlal		186651	22690	19987	19874

The descriptive statistics of our posts information can seen in Figure 1 and Figure 2. Even though there is noticeable variance in each month, the distribution of posts number and news article number does not of are basically the same so we can preclude the possibility of potential bias of missing dates and text information of news articles.

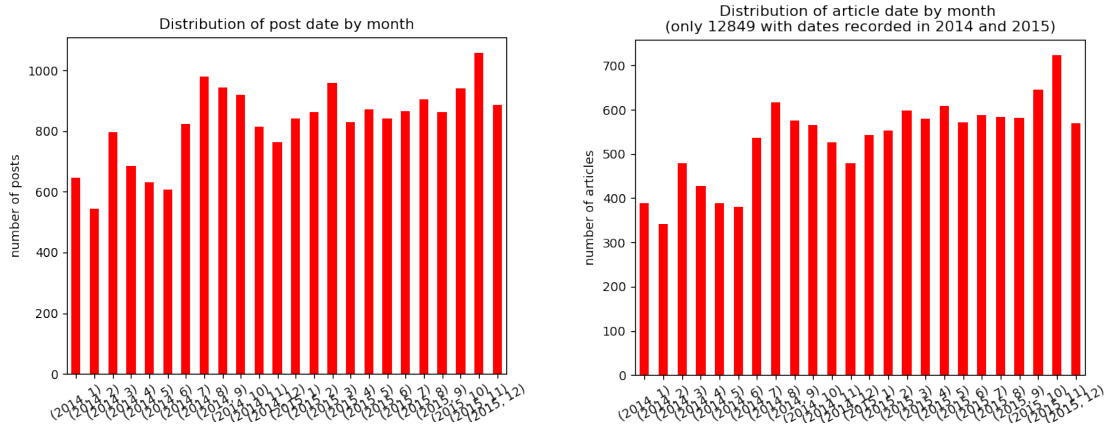

Figure 1. Distribution of number of posts and number of articles in each month

2. Features

Comment structure The first part of our predictive model is the comment information. By using the unique id of each comment under each post of news article, I can build a network structure of different users participating in the discussion of each post. When a user comment on other people's comment or post, I draw a directed edge from the sender to the recipient. For each post, I can get information of the comment network including basic attributes such as number of nodes (unique participants) and number of edges as well as more complex structural attributes to measure different patterns of the communication networks of news with different popularity. In order to measure how close each nodes are in the network, average clustering coefficient, transitivity, average shortest path are calculated, whether a network is strongly or weakly connected, minimum proportion of nodes and edges to be cut to get the network disconnected. I also calculate the proportion of periphery and center in the network to see how many nodes are isolated and focused in discussion.

Roles of important actors I also look at the attributes of active users in the post discussions. By active users, I mean those with high "karma". "Karma" is a metrics used by reddit to signal the activeness of users. There are two kinds of "karma" represented on the profile page of each user, "post karma" and "comment karma", which measures the users' rate of participation in initiating posts and making comments on other posts. Except for looking at the author's "karma" of each post, I also analyze the distribution of participants' "karma". Then I pick the author and ten percent of users from each post discussion with highest "karma" and look at how important and active they are in the discussion. I calculate the degree centrality, closeness centrality and betweenness centrality of these users to measure whether they take an active role in the discussion. In addition, I also calculate the hub and authority score of these users to verify whether they take the role of opinion leader in the discussion. In this way, I can examine whether the participance of opinion leader within the process of news diffusion in online platform can influence its popularity.

Text features The last group of the predictors I inspect is related the contents of news articles. I basically study the sentiment and content features of news articles. As for sentiment, I look at both the sentiment of headline and article body of each post. I categorize the sentiment of each post into eight sentiment scale from extreme negative to extreme positive. I refer to the sentiment detection algorithm Sentistrength proposed by Thelwall and his partners (2010) to predict the sentiment of each post's article and give them labels of sentiment categories for both headline and text body (if this method failed I will use SVM and NaiveBayes Classifier instead).

Regarding the content feature, I will look at both and macro and micro level of the body. On one hand, in the macro level, I obtain nine topics of the news article corpus using Gensim's LDA (Latent Dirichlet Allocation) model to perform topic modeling algorithm. After getting the nine topics of the news corpus, I categorize each article based on its topic distribution of each topics. On the other, in the micro level, I study the general summarizing features of each article's text body, including word count, average length of each sentence, average word length and distribution of different part of speech. Considering most of study on predicting online news popularity so far mostly look at the effect from news content, only when I control the most important features in text level, I can see whether comment structure and roles of opinion leaders can corporately impact the diffusion of news article.

Model I build predictive model on both number of comments and upvotes of each post. It is hard to define popularity of information, which can be either the participance rate or the willingness of support by the public. And this two concepts are not necessarily equivalent in every contexts. The summary statistics of number of comments and upvotes are shown in Table 2 and Figure 2. Though there's unknown reason why there are still posts with upvote less than 50 after I filter those below 50 in the preprocessing stage, it might not influence the result for there are only 246 posts below 50.

Table 2. Summary statistics of number of comments and upvotes (n = 19874)

	maximum	minimum	mean	median	std.	.25 quantile	.75 quantile
# comment	27896	0	317	80	767	25	268
# upvotes	106049	0	1544	283	4250	99	976

We can see from the table that most of the posts have very small number number of comments and upvotes. Only a small proportion of posts have very high number of comments and upvotes but the range of maximum and minimum number are very large. The extremely large values of upvotes tend to scatter much farther away than those larger values of comments. And the two variables are not strictly linearly correlated. We can see the detailed distribution of and correlation between number of comments and upvotes from Figure 2. and Figure 3.

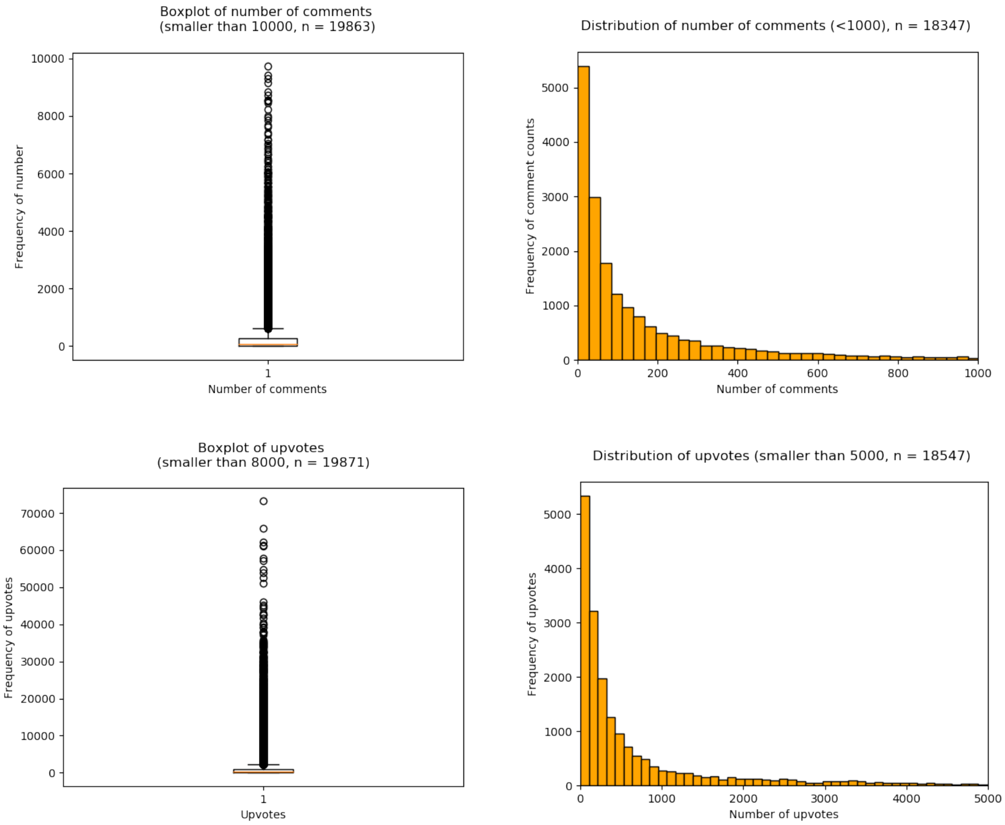

Figure 2. Boxplots and histograms of number of comments and upvotes (n = 19874)

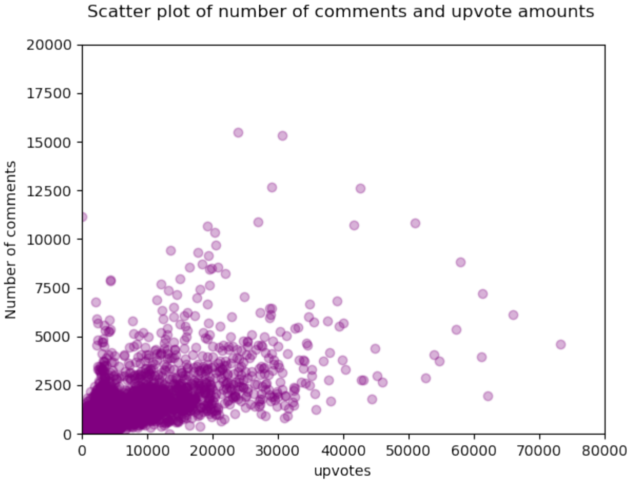

Figure 3. Scatter plot of sample number of upvotes and comments (n = 19847)

I use lasso regression to build the predictive model. Since there are more than twenty of predictors in the model and the sample size is a bit large, it would be a better choice to utilize linear method to build a model which tend to gain more generalization power to even larger data sets. In addition, by using lasso regression, I can shrinkage the coefficient of predictors with negligible impact to zero and identify those predictors which are of dominating importance.

After normalizing all the variable I tune the model by using different regularization parameter from 0.1 to 50 with 0.1 interval. I look at the R square and mean square to find the best parameter alpha for the model. Then I evaluate the model with the performance of dummy regressor and model only include content features.