

The Automated Script Reviewer

Katherine Schinkel
University of Virginia
Data Science Institute
kms6bn@virginia.edu

Marcus Rosti
University of Virginia
Data Science Institute
mer3ef@virginia.edu

ABSTRACT

This paper discusses the application of text mining techniques and regression analysis to two 90s sitcoms as a means to predict the shows' viewership and rating. We collected ratings from IMDb and viewership from Wikipedia's reported Nielsen ratings. We use tf-idf as predictors to our regression model.

Keywords

Text Mining; Regression Analysis; Machine Learning; Automated Systems

1. INTRODUCTION

TV shows cost an enormous amount of money. For instance just to film and develop a pilot, Priceconomics says it costs \$2 million to develop a typically 30 minute comedy or over \$5 million to make an hour long drama and even that is a low estimate.¹ That's a substantial investment into a project that may never be seen by the public. In the same article they say 98% of scripts fail to be made into pilots and of those, around half get a full season and an even smaller percentage make it past one season.

This creative and expensive process touches a number of people starting from pitching ideas for shows to actual filming the season. The step we focus on has to do with the script screening and development process. A production company will field many ideas and make a call for scripts based on those ideas. A producer must then screen through scripts and rate them only on their intuition or even some cognitive bias towards the writer. A producers time is valuable but on top of that the chosen script must maximize the chance of developing into a hit show.

Our research addresses automating the script review process or at least providing assistance to the reviewer via an automated system. We apply the text mining ideas of text

preprocessing and feature engineering to predict the continuous values of IMDb rating and Nielsen viewership. To motivate those two values, a producer cares about two things when developing a show. On the less import end (monetarily wise), the show should be a high quality show, so the script needs to be well written and should receive a high IMDb value when it airs. But more importantly the show needs to be viewed by many millions people to drive advertising or generate more subscriptions in the case of HBO or Netflix. Thus again, our model needs to be able to predict with some certainty these values.

We used two 90s Sitcoms, both being widely viewed and rated, to at least validate that our method could accurately model that response. Given that these shows have proliferated that culture we expect to see at least come predictive power to our approach and conclude that there is a correlation between the script and the audience response.

2. LITERATURE REVIEW

Researchers have explored sentiment analysis of text thoroughly. Bo Pang and Lillian Lee explored a multiclass approach to categorizing movie reviews as either positive, average and negative. They used support vector machines and a one versus all separation in order to classify them [1]. That research expanded a traditional binary classification into a multivalued problem.

In a separate field, Saxena explored applying a data mining techniques to screen resumes as part of a job application. In this case, the author explains that as the internet has become more popular, people are able to apply to many more jobs thus online job postings become flooded with resumes. To parse through these, they should apply several information retrieval techniques but this paper focused on skills extraction, finding unique features and dealing with the specific setting of the resume. They had mixed results and concluded that more research would be required to find a more confident parser [2].

Lastly, Pivotal, a data sciences research company, attempted a script reviewer with success. They extracted information from scripts that included speaker and scene delegation, for which they did extensive feature engineering. For example, they were able to distill the personalities of speakers via topic modeling and that feature engineering and include this with their linear model [3]. However, the authors did not reveal more intimate details of their modeling and approach.

3. METHOD OVERVIEW

¹<http://priceconomics.com/the-economics-of-a-hit-tv-show/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CS Text Mining Spring '16

© 2016 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

Our model follows a standard pipelined approach and employs 10 fold cross validation to justify the results.

3.1 Data Parsing

To clean the text, we followed a standard text mining pipeline, stemming, tokenizing and removing stop words. This removes any words that are too frequent to be meaningful in predicting the outcome of episodes. It also combines the forms of words that could be used in different moods and tenses.

The two response variables we used were again the IMDb mean rating for the episode and the Nielsen rating for viewership rating. For the IMDb rating, we paired the episode with its corresponding rating. The rating of the episode should stand on its own since it could come days weeks or years after the actual episode airs. Our claim here is that the rating is independent of the timing of the episodes air. However, for the Nielsen rating, it would not make sense to pair the episodes viewership with the actual episode. Consider someone watching the show. That person has already come to the decision to watch the show independent of the script because it is unfolding before them. We, instead, map the next episodes viewership with that episode; moreover, $Script_i$ corresponds to $Viewership_{i+1}$. We model it that way because if a person were to watch an episode, enjoy watching it because of the script and then be more likely to watch the next one. In this case, we capture the temporal aspect of the airing.

3.2 Regression Inputs

We explored two options as inputs to our regression model. One TF-IDF and one topic model. We use input

3.2.1 TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) is a way of weighting tokens in language model so as to highlight import words. It weights the number of times a word appears in a document (the Term Frequency) by the inverse of the number of documents that word appears in all documents in the training set. So two examples, a stop word like 'a' will appear in every single document. A typical IDF metric will force that value to result in a 0 thus removing the term from our model.

Our input matrix to the regression equation is a document term matrix where the values in each cell are that weighted TF-IDF value.

3.2.2 Topic Modeling

Latent Dirichlet Allocation (LDA) is a generative language model used to formulate topics from a document corpus. For each document in the corpus, the LDA model provides a vector of probabilities representing how likely each topic applies to that document. This allows each document to contain multiple topics, mirroring the complexity of a single sitcom episode.

3.3 Model Validation

We followed a testing procedure based on 10 fold cross validation. In this procedure we compared our regression model, built on each training fold, with the mean of the response, again on the training fold. We use our model to predict the response on the test set and compare it to the mean of the response on the training set. Then using a two

sample t test to make the claim that our model exceeds the performance of the mean.

$$t^* = \frac{\bar{Y}_1 - \bar{Y}_2}{\sqrt{s_1^2/N_1 + s_2^2/N_2}} \quad (1)$$

4. RESULTS

Our null hypothesis again was that the model performed just as well as the mean and our alternate hypothesis is that the model outperforms the mean in terms of mean squared error.

Method	Target	Mean	Model	t value
TF-IDF	IMDb	0.134	0.175	-1.713
TF-IDF	Nielsen	50.018	70.269	-1.226
LDA	IMDb	0.133	0.132	0.062
LDA	Nielsen	45.858	44.872	0.096

5. ANALYSIS

Our model did not outperform the mean in any of the four tests by a statistical margin. In TF-IDF in fact underperformed the mean. Our topic modeling did out perform the mean by a small margin but so small that the t value is almost zero. This shows that, by these two inputs the topics of the episode and the words used in the episode, are no stronger than a naive approach of guessing the viewership and rating. Since these did not show a stronger correlation we do not recommend using these values as predictors to a regression model.

6. FUTURE WORK

Although our modeling results indicate that there may not be a relationship between television scripts and ratings, applying more extensive feature engineering could supplement our model. Our modeling was limited to unigrams from transcripts. Future work could explore incorporating bigrams or trigrams to capture important phrases within episodes. Additionally, Pivotal achieved success with extracting metadata from the transcripts, such as count of scenes in an episode and count of characters in a scene. Pivotal also extracted speaker characteristics and indicated that they were significant to their model.

Our model was trained on a small sample size of around 250 sitcom scripts. In the future, we would like supplement our model with a larger training set of documents. Training an LDA model with a larger dataset could result in more meaningful topics that could be significant to the linear model.

Future work could also include exploring other modeling techniques. Pivotal implemented an ElasticNet regression model to significant variables (indicated by a linear model applied to each variable). Alternatively, we could explore this topic as a classification problem by binning episodes into categories. In this case, we could apply SVM, a method that proved to be successful by Pang and Lee when classifying multiclass ratings [1].

7. REFERENCES

- [1] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual*

Meeting on Association for Computational Linguistics,
ACL '05, pages 115–124, Stroudsburg, PA, USA, 2005.
Association for Computational Linguistics.

- [2] C. Saxena. Enhancing productivity of recruitment process using data mining and text mining tools. 2011.
- [3] J. Vawdrey. Using data science to predict tv viewer behavior and formulate a hit tv show, 2015.