Predicting Online Review Helpfulness: Feature Selection Through Ablation

Ryan Mannion
Georgetown University
LING-472, Spring 2020
ram321@georgetown.edu

Abstract

Showing helpful reviews to users is predicted to have contributed around $2.7 billion USD to Amazon's media products sales in 2008 (Spool, 2009). Predicting the helpfulness score of a given review could allow smaller merchants to do just the same without having to rely on their smaller user base to provide judgements. This project uses Python to extract features from online food reviews and train multiple Logistic Regression models with varying feature sets to predict whether or not a given review is helpful. No model performed below the baseline zero rule. Additionally, no model outperformed all other models on all metrics, though the model which scored the best on accuracy and f1 achieved scores of 0.793 and 0.856 respectively.

1 Introduction

In 2016 the Pew Research Center estimated that roughly 8 in 10 Americans are online shoppers (Pew Research Center, 2016). As the number of online shoppers and retailers increases, it is more important than ever for retailers to get and keep the attention of shoppers. One way to do so is through the use of online reviews: seeing reviews on a website makes a shopper more confident in the legitimacy of the retailer, especially if those reviews are good. By presenting a viewer with high-quality reviews, retailers are able to show customers that other people are using the website, receiving their products, and – hopefully – having a good experience.

This process often takes place through the presentation of a question along the lines of "Was this review helpful?" to the user. Reviews with more helpfulness votes will be presented to other users first, as opposed to another ordering (e.g. chronological). This naturally puts smaller merchants at a disadvantage, as they will have fewer reviews and fewer helpfulness votes.

The purpose of the present project is to develop a model to predict the helpfulness of a product review based on linguistic features. Additionally, we would like to discover which features had a large impact on the model, and test multiple feature combinations to find the best model.

1.1 Previous Work

The topic of modeling online review helpfulness has been studied by numerous others, often on datasets pulled from TripAdvisor or other areas of Amazon's Marketplace (e.g. Books, Technology, etc.). Krishnamoorthy (2015) showed on data from tech reviews that verb classification into classes of 'state,' 'interpretive,' and 'descriptive' aided Random Forest model performance. Singh et al. (2017) found that measuring the amount of information produced with each word was a relevant feature, while lexical predictors such as stop words had less of an effect. Lee and Choeh (2014) showed the effectiveness of "Reviewer Characteristics," i.e. non-linguistic features, in their study of media sales on Amazon (books, DVDs, etc.).

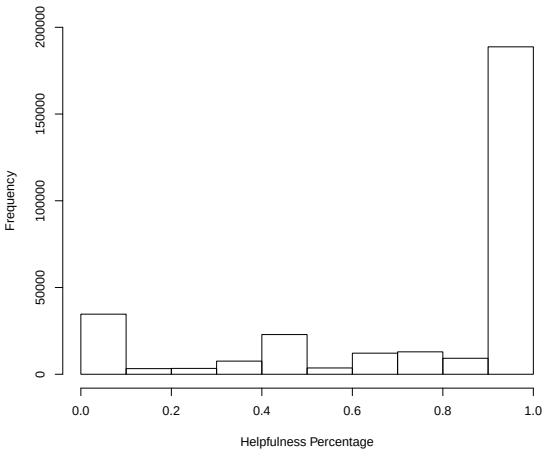

Figure 1: Histogram of helpfulness percentages (# marked helpful)/(Total votes) after removing reviews with no votes

Table 1: List of the 18 features included in each "feature group." Features are extracted from reviews using spaCy and Python

Simple Review (SR)	Simple Summary (SS)	Meta (M)	Review Readability (RR)
# sentences	# sentences	Star rating 1-5	Flesch-Kincaid Grade Level
# tokens	# tokens		Flesch-Kincaid Reading Ease
# characters	# characters		Dale Chall Index
Average tokens/ sentence	Average tokens/ sentence		SMOG
Average character/ token	Average character/ token		Coleman-Liau Index
			Automated Readability Index
			FORCAST

2 Methodology

2.1 Data

The data used in this project is the Amazon Fine Foods Reviews dataset, uploaded by the Stanford Network Analysis Project (SNAP) to kaggle.com in 2016.[1] The license for the data is designated as CC0: Public Domain, and is therefore available for free use. It includes 568454 reviews over 10 years until October 2012. Importantly, the dataset includes "HelpfulnessNumerator" and "HelpfulnessDenominator" columns, which specify the number of votes for *helpful* and the total number of votes, respectively. Figure 1 shows a histogram of helpfulness percentages after removing reviews with no votes. The dataset also includes information such as a product and reviewer ID, a timestamp, a star rating out of 5, and a summary of the review (user-provided, similar to a subject line).

2.2 Hyperparameters & Preprocessing

The data was first shuffled and split into three sets: (1) a train set, (2) a development-test (dev-test) set, and (3) a test set at 80%, 10%, and 10% respectively. The train and dev-test sets are used for model training and hyperparameter tuning before final testing on the test set. The sets were then processed by the feature extractor (see §2.3). Prior to model training, the data is filtered by two hyperparameters: `minimum_votes` and `help_boundary`. The parameter `minimum_votes` determines the minimum number of votes a review needs to have to be included in the experiment, and `help_boundary` is the decision point for whether or not a review is classified as 'helpful.' In this experiment, `minimum_votes` is set to 15, and `help_boundary` is set to 75% (i.e. a review is considered 'helpful' if 60% or more voters mark it as such). Previous studies have used both 60% (Ghose and Ipeirotis, 2011) and 75% (O'Mahony and Smyth, 2010). We opted for 75% as an arbitrary cutoff with the intuition that helpful reviews displayed to a shopper should be very clearly helpful. Should a review meet both hyperparameter criteria, its features are entered into an array to be used in model training or testing.

2.3 Feature Extraction

Four primary categories of features totaling 18 features were explored as part of this project: (1) Simple Features for Review Texts (SR), (2) Simple Features for Summary Texts (SS), (3) Meta Features (M), and (4) Readability Features for Review Texts (RR).[2] The features in each feature group are listed in table 1. These features were chosen based on their inclusion in previous studies, as well as our ability to write the code necessary in the time granted for this project.

To extract these features, the python library `spaCy` was used.[3] `SpaCy` streamlines the language processing aspect of pre-processing, as it is easy to work with and comes with pretrained language models to inform its parsers. Additionally, `spaCy`'s 'universe' allows for the creation of pipelines with additional components like `spacy-readability`, which provides the tools to calculate the various readability scores. [4]

2.4 Model Selection & Ablation

The model we selected is a logistic regression model from the python library `scikit-learn`.[5] The model is trained with a feature array of length

[1] https://kaggle.com/snap/amazon-fine-food-reviews

[2] Explaining the idea behind and calculation of each readability metric is outside the scope of this short paper. For a detailed look at readability metrics, see (DuBay, 2004).

[3] https://spacy.io

[4] https://spacy.io/universe/project/spacy_readability

[5] https://scikit-learn.org

n and a corresponding array of length n with the binary labels for helpfulness as decided by the `help_boundary` hyperparameter. The trained model can then be given an array of unseen features (e.g. from the test or dev_test sets) and predict whether or not the review represented by that specific feature vector is helpful.

To investigate which features have an strong impact on the performance of the model, we perform a feature ablation with five conditions:

Condition	Features
Condition 1	SR
Condition 2	SR & RR
Condition 3	SR & M
Condition 4	SR & SS
Condition 5	SR, M & RR
Condition 6	SR, SS & RR
Condition 7	SR, SS & M
Condition 8	All Features

Table 2: Ablation conditions for features extracted from review data. Note: Simple Review Features (SR), Simple Summary Features (SS), Meta Features (M), Review Readability Features(RR)

Since the purpose of the present project is to model the helpfulness of reviews, each condition contains at least the basic information about the review (SR). Additional feature groups are added to observe their impact on the model's predictions as measured by Accuracy, Precision, Recall, and F1 score.

3 Results

With `minimum_votes` set to 15 and `help_cutoff` set to 0.75 there were 10,454 train and 1,208 test reviews. The most common tag in the train set on a binary scale for helpfulness (0: not helpful, 1: helpful) was 1. In table 3, the baseline condition is the 'zero rule,' which marks the condition where the model predicts the most common tag for each test review.

No condition tested scored below the baseline, though conditions 1 and 2 both predicted the zero rule, and as such achieved the same score. Additionally, no condition performed the best on every metric. Model 7 (SR, SS, M) achieved the highest accuracy and f1 scores.

Condition	Acc.	Prec.	Recall	F1
Baseline	0.722	0.722	**1.000**	0.838
1	0.722	0.722	**1.000**	0.838
2	0.722	0.722	**1.000**	0.838
3	0.791	**0.862**	0.846	0.854
4	0.725	0.724	**1.000**	0.840
5	0.782	0.858	0.837	0.847
6	0.725	0.724	**1.000**	0.840
7	**0.793**	0.859	0.853	**0.856**
8	0.791	**0.862**	0.846	0.854

Table 3: Ablation results for the 8 test conditions. Zero Rule is the score when the most common tag is predicted every time. Conditions 1 and 2 predicted the zero rule, and conditions 4 and 6 likewise scored the same. Condition 7 had the highest accuracy and f1. No condition scored worse then the baseline. (Acc: Accuracy, Prec: Precision)

Readability features did not seem to provide any benefit to the model: condition 3 (SR, M) scored higher than condition 5 (SR, M, RR) on each metric. Similarly, condition 4 (SR, SS) and condition 6 (SR, SS, RR) scored exactly the same. Conversely, the relatively high scores by the models which include the meta features (3, 5, 7) seem to indicate the predictive power of that feature group. Using R to calculate Pearson's correlation, the star rating of a review and helpfulness have a positive correlation (r=0.391). The breakdown of review ratings by their binary helpfulness is shown in figure 2:

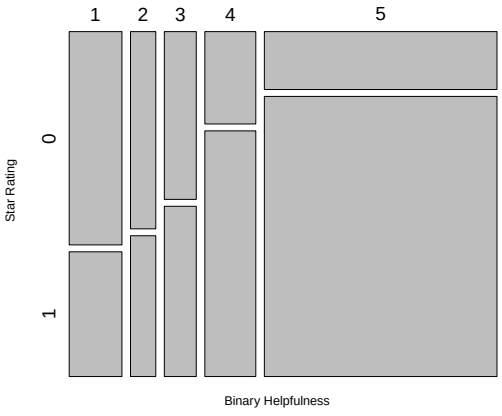

Figure 2: Breakdown of review ratings by binary helpfulness. r=.391

4 Discussion

This project has shown that it is possible to model review helpfulness based on linguistic features. Surprisingly, the readability of a review showed

little effect. Based on intuition, we predicted that readability would have a larger effect: reviews which provide information clearly are often more useful than those which ramble incoherently. This could have to do with the calculation of readability features being primarily based on the use of long or uncommon words, which might not be the case for many online reviews. Readability might be a metric better suited for a study on data akin to that of academic articles, which are known to use a lot of jargon.

The lack of features relating to the sentiment of a review is due to time constraints, and is a viable avenue of future study. Additionally, a different dataset which provides more information about the reviewers as well as the products themselves could provide useful information about, for example, how often that particular user writes reviews or the average star rating of the product. These non-linguistic and meta features could prove to be useful for model performance.

Additional feature engineering could reveal more information about the makeup of a helpful review, though for the purposes of this project we are satisfied with the results.

References

William H DuBay. 2004. The principles of readability. *Online Submission*.

Anindya Ghose and Panagiotis G. Ipeirotis. 2011. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1498–1512.

Srikumar Krishnamoorthy. 2015. Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, 42(7):3751–3759.

Sangjae Lee and Joon Yeon Choeh. 2014. Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046.

M. P. O'Mahony and B. Smyth. 2010. A classification-based review recommender. *Knowledge-Based Systems*, 23(4):323–329.

Pew Research Center. 2016. Online shopping and e-commerce.

Jyoti Prakash Singh, Seda Irani, Nripendra P. Rana, Yogesh K. Dwivedi, Sunil Saumya, and Pradeep Kumar Roy. 2017. Predicting the "helpfulness" of online consumer reviews. *Journal of Business Research*, 70:346–355.

Jared Spool. 2009. The magic behind amazon's 2.7 billion dollar question.