Smart Review Model

Shaan Arora | Miles Child
Jack Krolik | Manav Sharma

Abstract:

Automated review response (ARR) software is a service offered natively by Google and Facebook. Their response software does not use NLP and requires manual input of an array of review responses that map to specific characteristics of a review (i.e., the number of stars given on the review or specific tags mentioned in reviews). Thus, current ARR systems do not offer the following benefits that we are seeking to provide:

1. Intelligent and user-specific response insights
2. Effective sentiment recognition and response

Aside from ARR systems, other researchers have created models that sift through reviews and denote keywords included in the response (or custom tags). Some datasets include reviews and an array of emotions as response variables to the review.

Additionally, current products do not offer a viable solution for personal, accurate review responses. This excerpt from widewail coherently describes the issue with current ARR products:

"Contrary to conventional wisdom, Google/Facebook review response automation doesn't scale as well as a managed online reputation management service because these solutions can't address negative or nuanced reviews, still leaving extra work for you."

Our goal is to create an MVP that offers the increased functionality that current products do not.

Introduction:

Over the past few years, the popularity of online reviews has risen exponentially and it is clear that customers place great importance on these reviews. According to Dixa, "93% of customers will read online reviews before making a purchase." Furthermore, this dependence on reviews incentivizes businesses to solicit reviews to increase the review count. A product or business may be superior to its competitors, but if it has no reviews, it is unlikely that a consumer would choose it if better-reviewed products/businesses were available.

According to Google, businesses that respond to reviews are 1.7 times more trustworthy than companies that don't. Additionally, according to Brighlocal's Local Consumer Review Survey, 57% say they would be 'not very' or 'not at all likely to use a business that doesn't respond to reviews. Although the importance of review response is significant, 63% say that at least one company they reviewed never even responded. To make it even more challenging for businesses to deal with online reviews, according to ReviewTracker data, 53% of customers expect companies to respond to negative reviews within a week.

In comparison, 33% have a shorter time frame of 3 days or less. Therefore, to make businesses' lives easier, they need a way to respond to reviews efficiently without throwing more people at the problem. As such, our group has set out to solve this problem using natural language processing and various machine learning techniques to identify review sentiments and respond to them effectively. In doing so, businesses will have an automated solution to one of the most salient factors that impact the success of their business.

Methodology:

The Yelp reviews dataset consists of reviews from Yelp. It was extracted from the Yelp Dataset Challenge 2015 data which was found on Hugging Face. We will be using the test.csv from this dataset as although it is smaller than the train.csv it is still 50,000 rows which is plenty.

Through the EDA process, we reorganized the column headers, checked for missing data, and looked for errors in any of the reviews. We noticed the recurrence of \n in the reviews which was most likely supposed to be where a new line starts. As such, we removed every occurrence of \n as well as the recurrence of "\"". "\"" is used to signify a word that is in quotation marks within the review. Afterwards, we converted the data types to strings for the 'Review' column and integers for the 'Stars columns. Finally, the last part of the EDA was to ensure that the data that is left is clean and everything left can be used by GPT-3 to measure sentiment. We also are looking to respond to reviews with curated responses that hit on the reviews left by the user. For this, we believe that is important that any review that is less than 20 characters should be removed from the dataset. See below for a preview of the dataset after the EDA.

	Stars	Review
0	1	I got 'new' tires from them and within two wee...
1	1	Don't waste your time. We had two different p...
2	1	All I can say is the worst! We were the only 2...
3	1	I have been to this restaurant twice and was d...
4	1	Food was NOT GOOD at all! My husband & I ate h...

We first used the TextBlob library to gather all of the adjectives that exist in each review. Then, we took the 20 most popular adjectives from all of the reviews and then denoted whether or not a review has those in the dataframe.

Results & Evaluation:

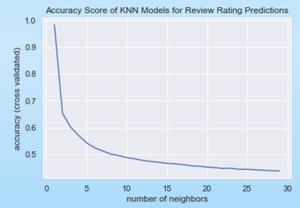
Initial Classifiers

Using the data frame of adjectives, we trained and hype-tuned with GridSearchCV a random forest, decision tree, and KNN classifier model. However, when first training and testing our model, we saw that the results were very poor.

Model	Hyper-tuned Best Parameters	Hyper-tuned Accuracy Score
Decision Tree Classifier	'max_depth': 6	32.69%
K Nearest Neighbors Classifier	'n_neighbors': 27	35.80%
Random Forest Classifier	'max_depth': 8 |'n_estimators': 110	36.17%

Classifier Using NLP

Although, the accuracies were still very low, and it was hard to predict the star given the classifier did not consider the sentiment of the review nor could it recognize the effect of a new adjective it had not seen. Therefore, we decided to use a different process using Vader, a natural language processing library, to gauge sentiment of the review. We then predicted the star of the review using KNN given the sentiment of the review. We chose KNN because it made more sense to use in the context of the problem and the hyper-tuned models performed very similar (only .37% disparity). With an accuracy score of 98.39% using the first nearest neighbor, the model performed extremely well. This means the review sentiment of the closest review was the most accurate prediction of the rating/star of the review. Here is a summary of the results below:



Model	Accuracy Score
K Nearest Neighbors Classifier	35.80%
Random Forest Classifier	36.17%
Decision Tree Classifier	32.69%
KNN using Vader Sentiment Library	98.29%

Generating GPT3 Review Responses

To generate responses to customer reviews, we extracted the adjectives from each review using TextBlob and NLTK. We then asked GPT-3 to "Generate a response to a customer review with the following adjectives: *(each review's adjectives)*." Then, we stored these responses in a new column of our data frame.

	Stars	Review	Response
0	1	I got 'new' tires from them and within two wee...	Thank you for your review. We apologize for yo...
2	1	All I can say is the worst! We were the only 2...	Thank you for your feedback. We are glad to he...
3	1	I have been to this restaurant twice and was d...	Hi there! I'm sorry to hear that you had such ...
4	1	Food was NOT GOOD at all! My husband & I ate h...	Thank you for your review. I am sorry to hear ...
5	3	This is a tiny Starbucks and it locations like...	We're glad to hear that you enjoy our little c...
6	2	Typical Starbucks coffee chain. 2 things I don...	Hi there,We're sorry to hear that you didn't h...

Impact:

The impact of our response generating solution is that it can save time for business owners or managers, who no longer must spend time manually responding to each review. This can be especially beneficial for businesses with a large number of reviews, as the tool can quickly and efficiently generate responses for all of them. Additionally, it can help the business maintain a positive reputation by providing timely and professional responses to customer reviews.

This can improve the overall perception of the business and potentially attract more customers. The tool can also be used to identify common issues or concerns raised by customers in their reviews. Given each response comes with a set of keywords or in our case adjectives, valuable insights can be shared with the business to improve its products or services and give priority to the most common customer concerns.

Conclusion:

Our original method of tagging key adjectives did not make for an effective model. This ineffectiveness could be because our reviews had diverse origins and were not based on a single product/business. On the other hand, our Vader-based model using sentiment analysis was much more accurate.

Our work with the GPT-3 model was very successful and allowed us to gain insight on the future of ARR systems. In the future, we would like to explore reviews centered around a single product or business. Also, it would be interesting to evaluate the impact of our generated reviews to see if they perform as effectively as manually written ones.

Related Work:

There is other related work that exists on HuggingFace that uses the dataset that we used in our project. The work that exists is mainly focused on training models and predicting star rating using the YelpReviewFull that we have used in our project. There is a model created by Shunian that is fed text, such as "I like you. I love you" and the model will return the percentages of the number of starts that would be denoted to this string: "LABEL_3: 0.554, LABEL_4: 0.356, LABEL_2: 0.086, LABEL_1: 0.003, LABEL_0: 0.001". There are 27 other models that are trained using YelpReviewFull that are also focused on predicting the star rating of a string.

References:

Libraries Used: GPT-3 (OpenAI), Pandas, TextBlob, NumPy, VADER NLP, Matplotlib, Sklearn, NLTK

Sources Used: https://www.widewail.com/, https://smallbusiness.withgoogle.com/free-google-training/how-to-respond-to-google-reviews/#!/, https://www.brightlocal.com/research/local-consumer-review-survey/, https://www.reviewtrackers.com/reports/online-reviews-survey/, https://www.reviewtrackers.com/reports/online-reviews-survey/, https://huggingface.co/datasets/yelp_review_full , https://huggingface.co/Shunian/yelp_review_rating_roberta_base?text=I+like+you.+I+love+you