Predicting “Writer” Compliments for Yelp Reviewers

Jacob Titherley, Jose Paulo Gonzales, Chih-Ning Yang

Introduction/Dataset

With the all the secret but not so secret perks of being an Yelp elite such as invitations to free parties, there is an incentive for Yelp reviewers to write good reviews. To be a Yelp elite, one of the requirements is to have quality reviews. There are multiple kinds of compliments a reviewer may receive for their reviews. Here, we seek to explore the characteristic of Yelp reviews and user profile that would allow a reviewer to receive “writer” compliments by using machine learning approach to predict the number of “writer” compliments a reviewer will receive for his/her reviews.

For this research, we used the dataset from the Yelp Dataset Challenge that contains 4.1M reviews by 1M users from 11 cities over 4 countries [1]:

- U.K.: Edinburgh
- Germany: Karlsruhe
- Canada: Montreal and Waterloo
- U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland

For our purpose, we made use of two of the json files provided by the Yelp Dataset Challenge:

- yelp_academic_dataset_review.json containing the following properties:
    - “review_id”: “encrypted review id”
    - "user_id":"encrypted user id"
    - "business_id":"encrypted business id"
    - "stars":star rating, rounded to half-stars
    - "date":"date formatted like 2009-12-19”
    - "text":"review text"
    - "useful":number of useful votes received
    - "funny":number of funny votes received
    - "cool": number of cool review votes received
    - "type": "review"
- yelp_academic_dataset_user.json containing the following properties:
    - "user_id":"encrypted user id"
    - "name":"first name"
    - "review_count":number of reviews
    - "yelping_since": date formatted like "2009-12-19"
    - "friends":["an array of encrypted ids of friends"]
    - "useful":"number of useful votes sent by the user"
    - "funny":"number of funny votes sent by the user"
    - "cool":"number of cool votes sent by the user"
    - "fans":"number of fans the user has"
    - "elite":["an array of years the user was elite"]
    - "average_stars":floating point average like 4.31
    - "compliment_hot":number of hot compliments received by the user
    - "compliment_more":number of more compliments received by the user
    - "compliment_profile": number of profile compliments received by the user
    - "compliment_cute": number of cute compliments received by the user
    - "compliment_list": number of list compliments received by the user
    - "compliment_note": number of note compliments received by the user
    - "compliment_plain": number of plain compliments received by the user
    - "compliment_cool": number of cool compliments received by the user
    - "compliment_funny": number of funny compliments received by the user
    - "compliment_writer": number of writer compliments received by the user
    - "compliment_photos": number of photo compliments received by the user
    - "type":"user"

However, for computation purpose and for preliminary exploration of the data, we used just 50,000 reviews for the training set and another 50,000 for the validation set.

Predictive Task

The aim of our research is to predict the number of “writer” compliments a Yelp reviewer will have using the least square linear regression model. Our main task is to find a way to populate our feature vector for the predictive task. To evaluate our model, we calculated the mean absolute error (MAE) as a way to quantify the accuracy of our model. The MAE is calculated using the following formula:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i|\,.$$

Where

$$AE = |e_i| = |y_i - \hat{y}_i|$$

$$Actual = y_i$$

$$Predicted = \hat{y}_i$$

The higher the MAE, the worse our model is for the predictive task. Likewise, if the MAE of our model is low, we can be relatively confident that our model is an appropriate fit for the current task. Therefore, MAE is what we used to validate our model’s predictions.

A baseline to serve as a control to compare our model to is a model using least square linear regression with the feature vector [1, number of fans for the reviews’ user_id]. We reason that the number of “writer” compliments a reviewer will receive is at least proportional to the number of fans the reviewer has who appreciate the way he/she writes.

The reason why we chose to use linear regression for our predictive task is that the prediction is able to take the form of a number in a wide range from 0 to as high as 2074. Therefore, a least squared linear regression is the better model to start with as opposed to other models that aim to perform categorization tasks.

The features we would like to explore and consider including in the linear regression feature vector are whether the reviewer was ever an elite, the number of fans the reviewer has, the reviewer’s average rating across all his/her reviews, and textual analysis of the review texts themselves.

Model

For our model we tried a few different things. For the feature vector our first instinct was to use a semantic analysis of a user’s reviews. We built a few different models using this kind of analysis; after taking the top 1000 non-stopword unigrams, top 1000 bigrams, we tried building our feature vector from combinations of one-hot encodings representing whether or not a unigram or bigram was represented in that user's’ reviews. Using unigram only, bigram only, and a combination of the two, our error was high, giving an MAE of about 55.6 at its lowest. We then tried to use the tfidf values of each top unigram in the review data, and that gave a high error as well, with the lowest being 36.7. After a few permutations of unigrams, bigrams, and tfidf values we determined that, oddly, the words that a user uses in their reviews is not necessarily indicative of receiving many compliments, at least with how we were approaching it.

The model that we ended up with was to use the average stars the user gave, the number of fans they had, whether or not they had ever been an elite, and the percent of their words that are stopwords. Our reasoning was:

- User’s average star ratings: is the user a critical, or a lenient reviewer? Or perhaps they’re somewhere in between. This “balance” could indicate that the user provides constructive criticism, not just harshness or leniency.
- Number of fans: fans are people who appreciate the writing style or content of a user’s reviews; the group of fans are the people most likely to compliment the user.

- Elite status: if a user is or has ever been an elite, they are likely to put a lot of effort and thought into their reviews.
- Percent of words that are stopwords: somewhat unintuitively, reviews that have a lot of stop words are easier to read and understand.

This model ended up being a lot simpler than what we had initially considered, but it goes to show that a more complex model does not necessarily fit the data better. The strength of this model versus previous ones is that its features provide a lot of information about the user; a user's elite status provides a lot of information about the user, as does the number of fans they have and the vapidness of their writing style. There are only a few features, but they vary a lot among users so they easily distinguish users. Our earlier models were weaker perhaps because they were based on facts about reviews that are likely to be similar among reviews and users; reviews for restaurants, for example, often have the same buzzwords in them and don't necessarily vary greatly from each other in quality based on the words themselves. Our final model's simplicity resulting in such a good MAE makes us wary of it being "too good to be true" and perhaps we're missing something. Due to the generalizability of a review's text, we feel that not considering the review text further, besides the percentage of stopwords, is a weakness in our model. A grammatical analysis of a review could prove fruitful in the future, but the bag-of-words approach simply does not provide enough distinction among reviews in similar categories.

Literature

The dataset we used was provided by the Yelp Dataset Challenge to encourage academic data science research to discover interesting and even useful trends in Yelp reviews. Because the competition has been ongoing for multiple rounds and for a few years, extensive research in all directions have been done for this set of data using machine learning methods. Although there is no existing effort made to also predict the number of "writer" compliments like our predictive task, other research groups use generally similar methods like the ones we used for their own prediction goals.

One winning group from UCSD used methods similar to what we tried in analyzing the review texts in their paper "Oversampling with Bigram Multinomial Naive Bayes to Predict Yelp Review Star Classes" [2]. In that paper, the students took on the challenge of predicting a reviewer's star rating based on the reviews they left. For that research project, the authors focused mainly on textual analysis. Their method involved finding top unigrams, bigrams, trigrams, and combination of these to identify the kind of words characteristic of different star categories. Some other research such as that done by a group of Stanford student in their paper "Predicting Usefulness of Yelp Reviews" [3] used various models we learned in class to categorize and make prediction for their dataset, with linear regression, Naive Bayes, Logistic Regression, and Support Vector Machine (SVM), showing that these are all possible models for this dataset although some work better and are more appropriate for specific prediction goals. For their purpose, they saw that logistic regression and SVM algorithms work better for categorizing a review as either useful or not.

For our purpose of not categorization but numeric prediction, least squares linear regression model is used, which is supported by the research done by another group of Stanford students in their paper "Applications of Machine Learning to Predict Yelp Ratings" [4]. In one section of their paper, they ran the regression model on feature subsets to determine which features are more indicative of business performance on Yelp. From their finding, it appears again that SVM performs better by yielding better accuracies than other models such as regression. It may appear from these paper that perhaps SVM produce better prediction for the Yelp dataset than linear regression. However, for our purpose, linear regression is more useful because what we seek to predict (number of "writer" compliments a reviewer has) is difficult to be divided into categories due to the wide range of such number.

Results and Conclusion

The best accuracy that we were able to achieve was an MAE of 2.6, meaning that on average our prediction was off by 2.6 votes. The mean absolute errors on the test data of each of our approaches were:

Technique	**Test MAE**
Bag-of-words unigrams	56.5
Bigrams	54.3
Bigrams and unigrams	46.9
tfidf	36.7
Tfidf and unigrams	36.8
Tfidf and bigrams	39.7
Tfidf and unigrams and bigrams	43.2
Number of fans (baseline)	13.2
(Shuffled) Stopwords percentage, elite status, num of fans, avg user ratings	3.61
Stopwords percentage, elite status, number of fans, average user ratings	2.6

Clearly, the bag-of-words approach was not a useful approach. Our theory is that reviews of the same category tend to have very similar language: the words themselves are not a good indication of whether a writing style will be well liked; rather the simplicity of the review is. This also is assuming a linear model; it is possible that there is a relationship between the words in the reviews and the compliments a user receives but just that it is not a strictly linear relationship.

The best features by far were the number of fans and stop words percentage. They got the MAE down below 10, and adding the elite status pushed it down further to our best score of 2.6. This best score was our test MAE, and its training MAE was 5.6, so we also tried to shuffle the training and test data. This resulted in a training MAE of 3.65 and a test MAE of 3.61. The baseline of 13.2 using only number of fans is a significant indicator of a good way to approach this problem. After extensively testing using lexical analysis (tfidf, bag or words), the significant jump from the lowest Tfidf result (36.7) to the baseline could mean that such lexical analyses are not a good influence for this problem, that we needed to use a better feature representation, or that we could have used a better model for prediction when using lexical analysis. At the other side of this, the baseline result and the subsequent features that improved it to 2.6/3.6 suggests that the number of “compliment_writer” compliments that a user gets depends more on simpler features. So in the end, the best model was not the one that tried to quantify good writing based on the actual content of the reviews, but rather the additional features (meta features if you will) that represent the way that each user was connected and interacted with others. The model that focuses on behavior rather than content was more effective.

Reference

[1] Yelp. 2017. Yelp Dataset Challenge. Retrieved March 5, 2017. <https://www.yelp.com/dataset_challenge>

[2] Hung K and Qiu H. 2014. “Oversampling with Bigram Multinomial Naive Bayes to Predict Yelp Review Star Classes” <https://kevin11h.github.io/YelpDatasetChallengeDataScienceAndMachineLearningUCSD/>

[3] Liu X, Schoemaker M, and Zhang, N. 2014. “Predicting Usefulness of Yelp Reviews” <http://cs229.stanford.edu/proj2014/Xinyue%20Liu,%20Michel%20Schoemaker,%20Nan%20Zhang,Predicting%20Usefulness%20of%20Yelp%20Reviews.pdf>

[4] Carbon K, Fujii, K, and Veerina P. 2014. “Applications of Machine Learning to Predict Yelp Ratings” <http://cs229.stanford.edu/proj2014/Kyle%20Carbon,%20Kacyn%20Fujii,%20Prasanth%20Veerina,%20Applications%20Of%20Machine%20Learning%20To%20Predict%20Yelp%20Ratings.pdf>