Aspect-based Sentiment Analysis on Hotel Reviews

Yangyang Yu
Stanford University
yyu10@stanford.edu

Abstract

In this project we explored varieties of supervised machine learning methods for the purpose of sentiment analysis on TripAdvisor hotel reviews. We experimented and explored with the factors that affect accuracy of the predictions to develop a satisfying review analyzer. We focus on not only the overall opinions but also aspect based opinions including service, rooms, location, value, cleanliness, sleep quality and business service. As a result, we implemented an analyzer that is able to predict rating and polarity of reviews on individual aspects. The accuracy of our predictors reached 70% to 75% for star-rating and about 85% to 90% for polarity.

1. Introduction

Nowadays, people frequently use reviews on online communities to learn about others' opinions and to express their own opinions on business. While a rating can be a good indicator for the opinion, text reviews are usually more elaborative. At the same time, reading text reviews is also time-consuming. For this project, we propose using machine learning algorithms to help extract opinions from text reviews. More specifically, we would like to build an algorithm that is able to differentiate individual aspects of the review. Being able to analysis the individual aspects is especially valuable since a business can vary in different aspects and that users might have different preferences and priorities. The results of aspect-based semantic analysis can be used in a wide variety of applications, including building better-customized recommendation systems and generating more informative summaries. We can also use the results as features of the reviews for the purpose of further analysis.

2. Dataset Collection

For this project, we will use the TripAdvisor data set collected by Wang et al.[2] The data set consists of 235,793 text reviews along with star ratings for the overall service and seven individual aspects. The reviews are organized in groups of hotels. We use the star ratings as the ground truth of the user's sentiment for both learning guidelines and accuracy analysis guidelines. The review title and texts are combined as the object of analysis.

3. Approaches

3.1. Feature Extraction

We selected the bag of words model to represent our review texts. We experimented with several feature extraction configurations.

We first experimented with the classic text analysis pipeline, where after eliminating the stop words, the occurrence of each word in the review text is counted. Then the TF-IDF (term-frequency times inverse document-frequency) is calculated to put more emphasis on the less common, thus more interesting words.

During our analysis of some preliminary results, we realized that due to the limited topic of reviews, the words that express sentiments are actually very common across all documents. As a result, the TF-IDF transform might not be suitable for our application. So we experimented with pipelines without the TF-IDF transformation.

We also explore the effect of using n-grams. We experimented with both character-based n-gram and word based n-gram. Character-based n-gram is supposed to help us reduce the effect of mis-spelling while word-based n-gram would provide us more information through phrases. The extra information that is only carried by multi-word phrases includes the extend of an opinion, e.g. 'very good', negations, e.g. 'not good', and the aspect, e.g. 'good service'. Since using n-gram drastically increased our feature size, we also applied explicit max and mean document frequency constraints to limit the number of features.

3.2. Model

We experimented with two models for the problem. We modeled it as a multi-class classification problem and a regression problem. To model the problem as a multi-class classification problem, we consider the one to five star ratings as the five classes. To model the problem as a re-

gression problem, we consider the rating as a continuous variable that can take the value from one to five. With both methods, We treat each aspect as a separate problem. One challenge is that aspect-based classification inputs and guidelines are quite noisy in some sense. Since it is possible for a user to leave a star rating for a certain aspect without mentioning the aspect in the text. And even if they did, it could be only a small portion of the text. One measurement is to compare the accuracy of a predictor trained on the aspect star ratings with the accuracy of a predictor trained on the overall star ratings. In this way, we will be able to see if training on specific aspect guidelines is actually helpful for the predictor to understand better.

3.3. Algorithms

We applied the Multinomial Naive Bayes algorithm, the linear SVM algorithm and the linear regression algorithm respectively.

Preliminary results and observations discovered one issue with our training data. Since people give a higher star rating a lot more often than a lower star rating, our dataset is very skewed, with about 5% one-star ratings and about 40% five-star ratings. As a result, it is very important to balance the input training data. We experimented with three popular methods to deal with unbalanced inputs, class weight, sub-sample and over-sample.

The class weight method applies a weight to each class based on the sample distribution. The class weight is calculated by the inverse of the number of class samples in the dataset. The penalty of each sample is then weighted by the class weight. As a result, a class with fewer samples in our dataset will be emphasized more to compensate the fewer number of samples.

The over-sample method repeatedly sample the classes with lower distribution in the dataset until the same sample number is reached as the classes with higher distribution.

The sub-sample method sub-sample the classes with higher distribution, so that the number of samples used training match the number of samples in the classes with lower distribution.

4. Results

4.1. Evaluation Methods

To explore the effect of different data set sizes, we sampled the available data set into four sub sets of sizes 10,673, 30,050, 45,970 and 94,926. For validation purpose, we randomly sample one third of each subset as the test set and the rest two thirds of each subset as training set. We train our predictor using the training set and make predictions using the predictor on the test set.

We used two accuracy measurements, star-rating accuracy and polarity accuracy. For each test sample, a star rating of a certain aspect is generated. The prediction is then compared with the ground truth to determine the accuracy of the predictor, the accuracy is calculated by the number of mismatched predictions divided by the size of the set.

For the star-rating accuracy, under the multi-class classification model, a prediction is considered mismatched when the predicted rating is different than the user-given rating; Under the regression model, a prediction is considered mismatched when the predicted rating and the user-given rating has more than a 0.5 star difference.

For the polarity accuracy, under both models, the results are defined into three groups, positive, neutral, and negative. All ratings above 3 are considered positive, all ratings below 3 are considered negative, while all ratings at 3 are considered neutral.

In the following sections, we analyze the various factors we experimented with by examining the accuracy of the trained predictors.

4.2. Class Balance Method Comparison

The results in Table 1 are generated based on the same test and training set grouping, using the SVM algorithm with the same parameter. The overall ratings are set as the prediction goals.

Table 1. Accuracy of the predictors trained with different class balance techniques.

	Training Set Accuracy	Test Set Accuracy
No Balance	0.585	0.551
Class Weight	0.718	0.614
Over-sample	0.610	0.562
Sub-sample	0.720	0.581

As we can see, the class weight method works the best in our application.

4.3. Model and Algorithm Comparison

As discussed in Section 3, we decided to experiment with two different models (multi-class classification and regression) and three different algorithms (Naive Bayse, SVM and Linear Regression).

The results are shown in the figures below. Figure 1 shows the star rating accuracy and Figure 2 shows the polarity rating accuracy.

As we can see, as the dataset size becomes larger, the SVM algorithm tends to generate better results. Thus, we chose SVM as the learning algorithm for our application. At this point, the star-rating accuracy is about 50% to 60%, while the polarity accuracy is about 70% to 80%.

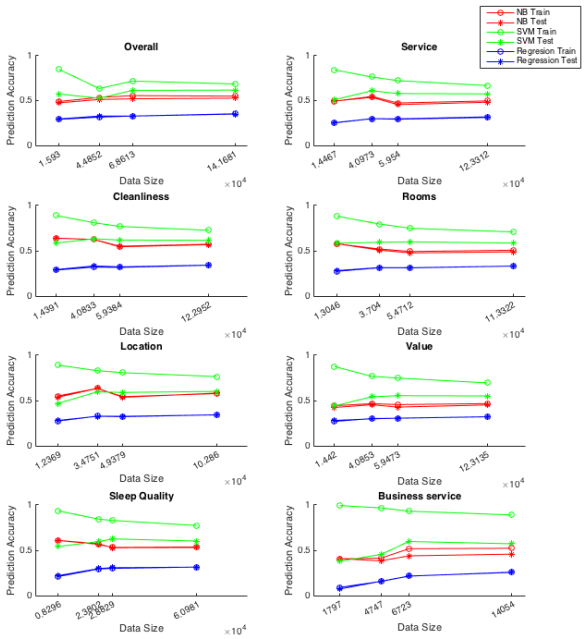

Figure 1. Star rating accuracy generated with various algorithms under various dataset sizes.

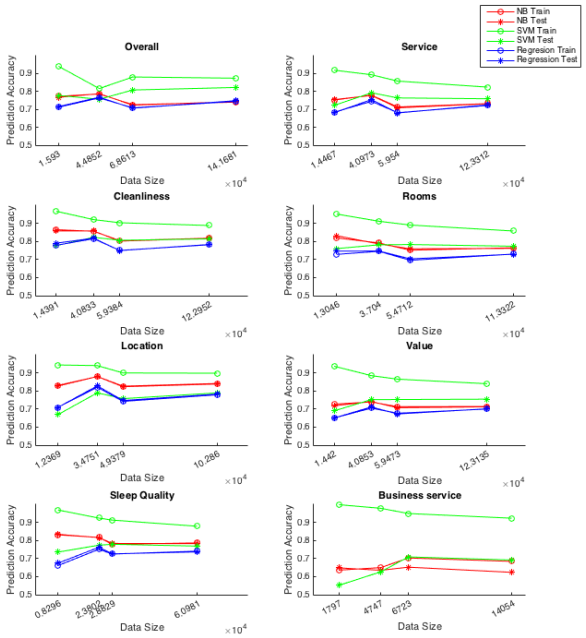

Figure 2. Polarity accuracy generated with various algorithms under various dataset sizes.

4.4. Feature Selection Comparison

As discussed in Section 3, we ran experiments with various n-gram configuration and pipelines with and without the TD-IDF transformation. Due to the limited computation resources, we only ran the experiments on the sub-dataset of size 45970. As we found in Section 4.3, the accuracy results generated on the sub-dataset of size 45970 and the accuracy results generated on the sub-dataset of 94926 are very close.

The exact n-gram configuration setups we used are given in Table 2. 'Min DF' and 'Max DF' indicate the constraints for the feature's document frequency. Any feature that appears too rare (lower than the 'Min DF' constraint) or too often (higher than the 'Max DF' constraint) is removed. For the 2/3-gram configuration, the 'Min DF' is set to 2 occurrences.

Table 2. Detailed n-gram configurations.

N-gram	Base	Stop-word Removal?	Min DF	Max DF
1	Word	Yes	None	None
5	Char	No	1%	90%
2 or 3	Word	No	2	20%

The results are shown in Figure 3. Both the star-rating accuracy and the polarity accuracy are given in the same graph.

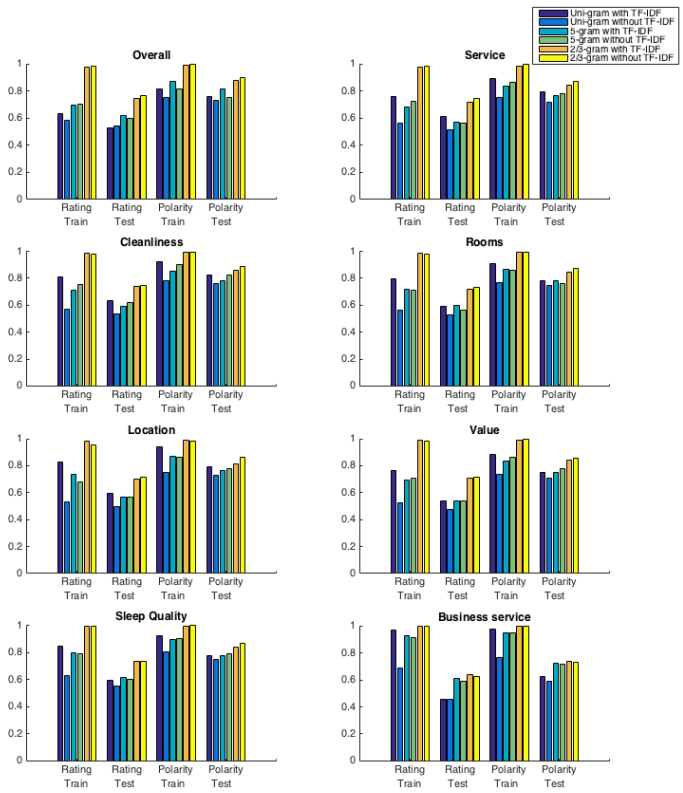

Figure 3. Star-rating accuracy and polarity accuracy generated with various feature configurations.

As we can see from the graph, feature selection plays a very important role in prediction accuracy. In general, word-based configurations work better than character based configurations. By selecting 2/3-gram with no TF-IDF transformation, we are able to improve the prediction accuracy significantly. The star-rating accuracy is now around 75% quite consistently while the polarity accuracy is between 85% and 90%.

Among all the aspects, the 'Business Service' accuracy is lower than the others. It is most likely caused by the insufficient training set size. Even though all the aspects are trained using the same reviews, a large portion of the reviews do not have a rating on the 'Business Service' aspect. Thus the training set for 'Business Service' aspect is much smaller than the other aspects. We believe the prediction accuracy will improve with a larger training set.

4.5. Overall vs. Aspect-specific Predictor

Lastly, we would like to examine the effectiveness of our aspect-specific predictors. For this purpose, we compared the accuracy of predictions generated by a predictor trained with only overall ratings and the accuracy of predictions generated by a predictor trained with aspect ratings. If our machine learning algorithm is indeed learning unique features for each aspect, we expect the aspect-based predictor to generate better results than an overall predictor. The results are shown in Figure 4.

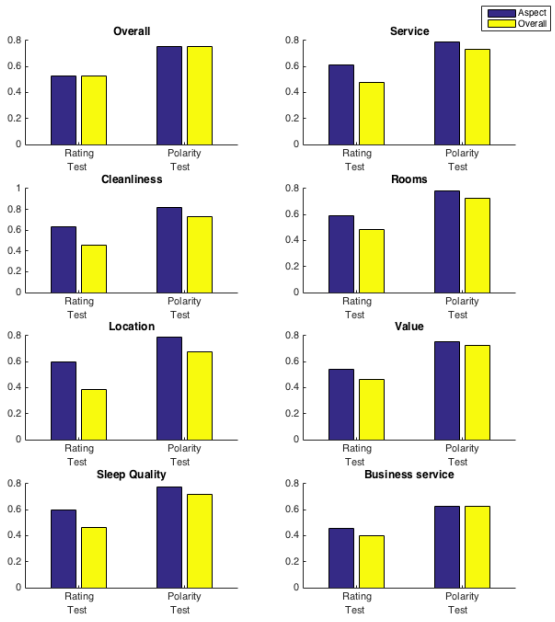

Figure 4. Aspect accuracy generated using an overall predictor and an aspect-specific predictor.

As we can see, the aspect training results are indeed better than the overall training results in all aspects, which means training the predictor using aspect-based objectives was able to learn the aspect-specific features.

5. Conclusion and Future Work

In this project, we examined various class balance techniques and data models. We achieved promising prediction accuracy on text review sentiments. Our prediction accuracy reached 70% to 75% for star-rating, and 85% to 90% for polarity. We abstracted the problem into a multi-class classification problem, selected the bag of words feature model using 2/3-word-long phrases, and executed the machine learning process with the SVM algorithm.

Among the design decisions, feature selection played a very important role in our design process, we were able to improve the prediction accuracy by about 15% selecting the appropriate features. Therefore, more optimization in feature selection will likely further improve the prediction accuracy. The bag of words model used in this project included full review text even when training for specific aspects. Therefore, the input data are effectively very noisy. Wang et al.[1] presented an aspect analysis method that could help separate the aspects and reduce the noise. It would very interesting to incorporate such aspect analysis into this project.

References

[1] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792, New York, NY, USA, 2010. ACM.

[2] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 618–626, New York, NY, USA, 2011. ACM.