Spam Detection in Hotel Reviews Report

Zhiyu Lin, Audrey Yuan, Yiran Liu, Yongzheng Chen

Introduction

As global networks rapidly develop, more and more industries are moving their businesses online. Customers can purchase products, book tickets and hotels, and even pick favorite restaurants through online applications. User-generated online reviews become one of the most important references for customers making purchase decisions. Unfortunately, among online reviews, a portion of them are deceptive. These reviews are deliberately written to sound authentic and misleads customers into making purchase decisions. In order to solve this problem, our team proposes to employ supervised machine learning algorithms to detect deceptive opinion spam, specifically we are interested in building a classifier that detects spam hotel reviews and separates those from truthful ones. Our team will build several machine learning classification models and evaluate the performance of each classifier.

Relevant (scientific) literature

The scientific literature we rely upon is the paper written by Tomas and Virginijus. The paper talks about the most popular machine learning models and classification methods that are used for text classification for example, Naive Bayes, Random Forest, Decision Tree, Logistic Regression, and etc. Since we are also doing a classification problem with our dataset, this paper offers a reference for us to choose the most appropriate models, and in our case, we will use Naive Bayes and logistic regression as our prediction models.

Data

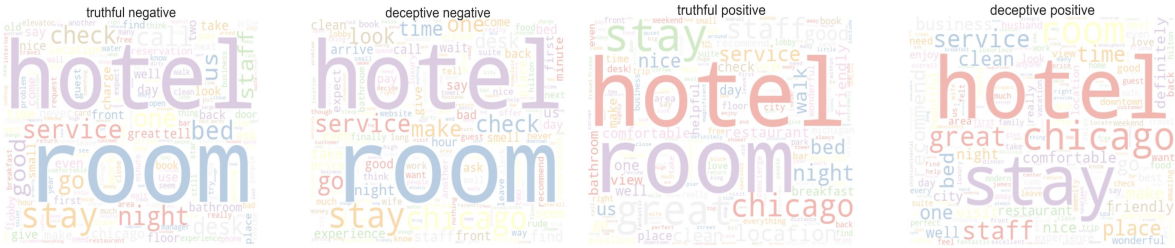
We have a corpus consists of truthful and deceptive hotel reviews of 20 Chicago hotels, the corpus contains 400 truthful positive reviews from TripAdvisor, 400 deceptive positive reviews from Mechanical Turk, 400 truthful negative reviews from Expedia, Hotels.com and Yelp.We build a Preprocessor class to clean and tokenize the reviews, removed all the punctuations in the corpus and the stopwords as well. The lemmatizer we used is WordNetLemmatizer, in the parameter of lemmatizer we put the results of the pos_tag in it, which are the token's text and token's tag, and TweetTokenizer as tokenizer.

EDA

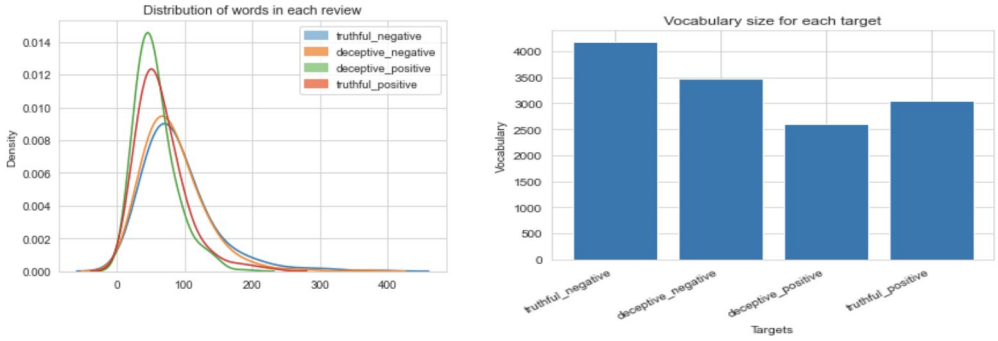
Training data

For the training data, we decided to control for positive or negative stance while doing EDA, just to see if the spam target is strongly correlated with the sentiment, or if there is any class imbalance. We

first plotted four word clouds for deceptive/truthful reviews cross-listed with positive/negative stances.



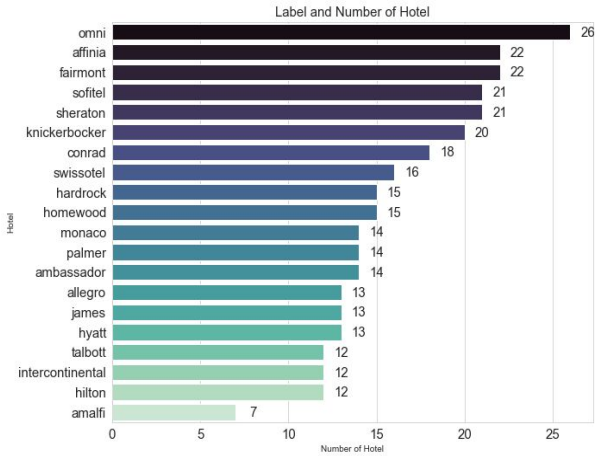
From the simple word clouds, we observe that all four categories feature “hotel” and “room”. However, there are some interesting words that stand out and could potentially be important features for prediction. For example, comparing the truthful positive and deceptive positive reviews, we see that deceptive reviews highlight “great” and “staff”, whereas these two words do not seem important in truthful reviews at all.



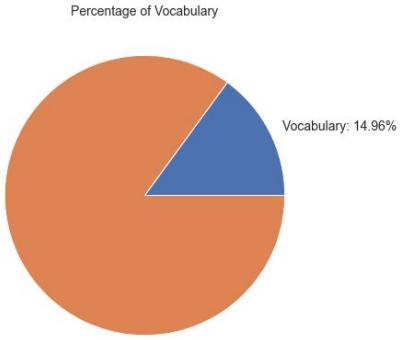
We also plotted the word distributions and vocabulary size for each of the four categories, shown above. In the density plot, we see that negative reviews are more left-skewed than positive reviews, meaning that negative reviews tend to be shorter than positive reviews. We do not observe significant differences in review length distribution between deceptive and truthful categories while controlling for stance. This gives us confidence that the model will be fitted on a balanced dataset. On the right hand side, we observe that the vocabulary sizes are roughly equal across categories while controlling for stance.

Testing data

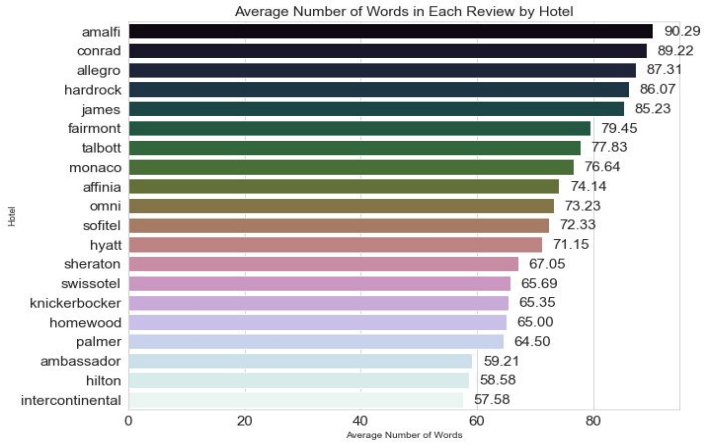
In the test data set, there are a total 320 reviews from 20 different hotels. The distribution of each hotel is shown in the “Label and Number of Hotel” chart. Omni is the most frequent hotel with 26 reviews and Amalfi is the least frequent hotel with 7 reviews.



There are 23373 tokens and 3497 distinct words in the test data set. Overall, 14.96% of tokens are vocabulary.



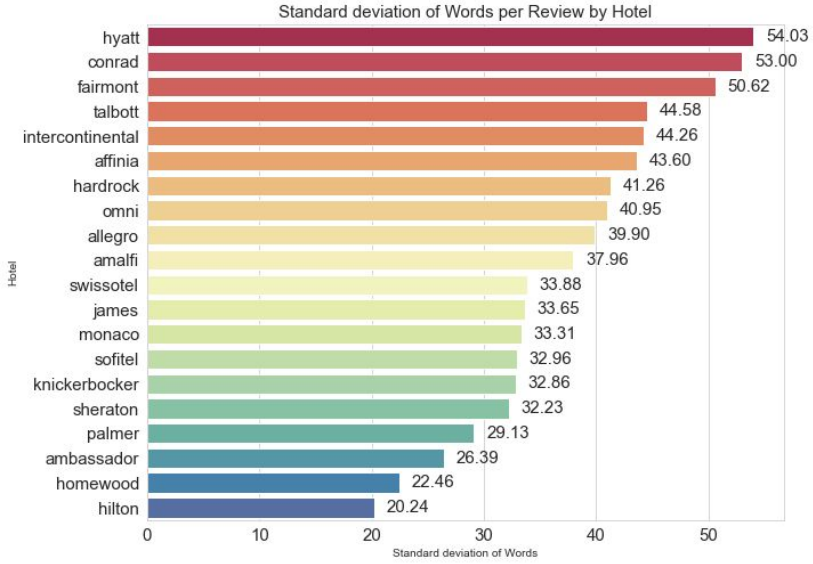
On average, there are 73.04 words in each review. In the "Average Number of Words in Each Review by Hotel" chart, Amalfi hotel has the highest average number of words in each review 90.29, and Intercontinental hotel has the lowest average number of words in each review 57.58.



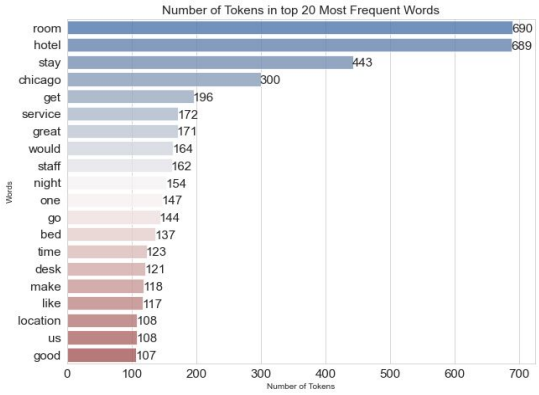
Standard deviation of words in each review is 38.79.

In the “Standard deviation of Words per Review by Hotel” chart, it is shown the standard deviation of words in each review by hotel. Hyatt hotel has the highest standard deviation of words in each review 54.03, and Hilton hotel has the lowest standard deviation of words in each review 20.24. This indicates that the number of words of review for Hilton are much clustered around the mean, and the number of words of review for Hyatt are much more spread out.

Standard deviations of words in test data set are large since the sample size is small, there are only 320 records for 20 hotels, number of reviews for each hotel range from 7 to 26. As the sample size increases, the standard deviation will decrease.

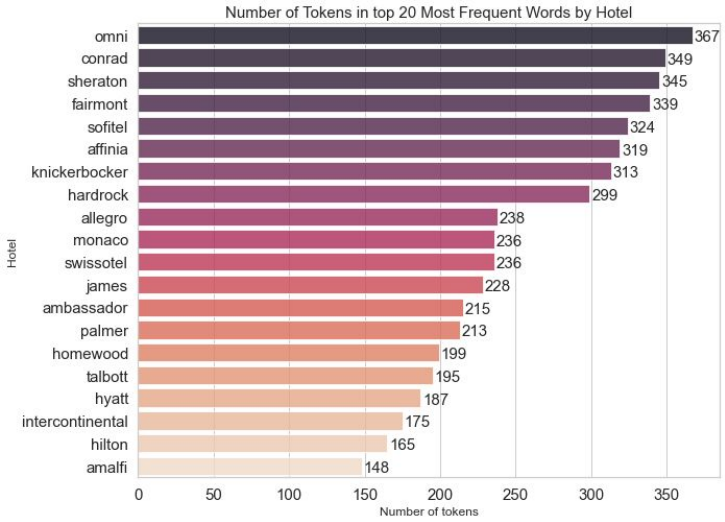


There are 4371 tokens corresponding to the top 20 most frequent words in the vocabulary. The “Number of Tokens in top 20 Most Frequent Words” chart shows the distribution of top 20 most frequent words, "room", "stay", "service", "staff", "desk" are very frequent words in reviews.



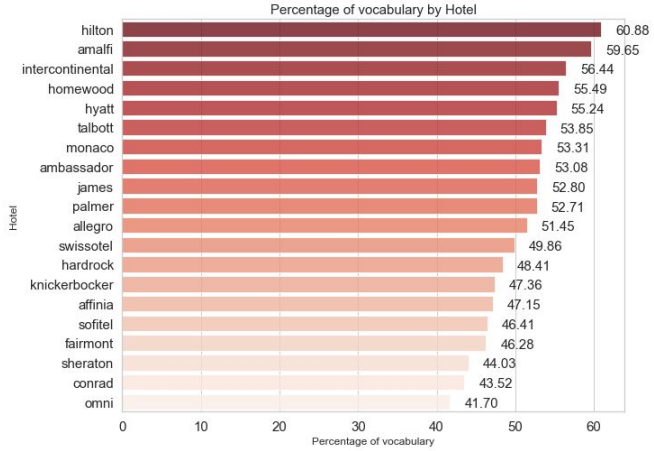
The chart “Number of Tokens in top 20 Most Frequent Words by Hotel” shows the distribution of

the number of tokens in top 20 most frequent words by hotel. Onmi hotel has the highest number of tokens in top 20 most frequent words. Amalfi hotel has the lowest number of tokens in the top 20 most frequent words.



The “Percentage of vocabulary by Hotel” charts show the percentage of vocabulary in each hotel. Hilton hotel has the highest percentage of vocabulary 60.88, Omni hotel has the lowest percentage of vocabulary 41.70.

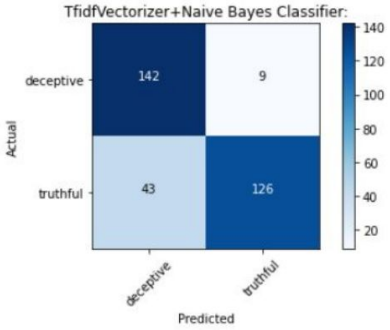
Percentages of vocabulary in 20 hotels are much higher than the overall percentages of vocabulary 14.96, one reason is that there are a lot of shared type (vocabulary) among 20 hotels, another reason is that the sample size of each hotel is small, as the sample size increasing, percentages of vocabulary by hotel will decrease.



Models

Naive Bayes

Naive Bayes method is a classifier based on bayes' theorem and independent assumption of characteristic conditions. For a given training data set, we first learn the joint probability distribution of input and output based on the independent assumption of feature conditions, and then, based on this model, we use Bayes' theorem to find the output y with the maximum posterior probability for a given input x. In this project, we used 'Authenticity' column as our label and TfidfVectorizer to transform the dataset into vectors, then we used train_test_split function in sklearn to split the dataset into train data and test data. After the model's prediction, we used accuracy, precision, recall and Macro F1 as the indicators of the performance of the model, and confusion matrix as well.

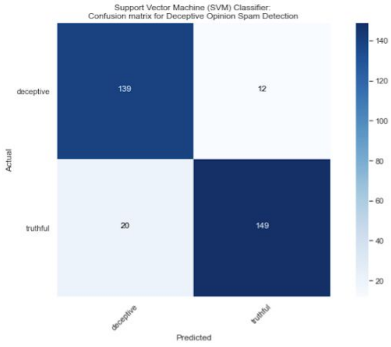


	precision	recall	f1-score	support
deceptive	0.74	0.97	0.84	157
truthful	0.96	0.67	0.79	163
accuracy			0.82	320
macro avg	0.85	0.82	0.81	320
weighted avg	0.85	0.82	0.81	320

Accuracy: 0.8375
Precision: 0.8504504504504504
Recall: 0.8429797405854462
Macro F1: 0.837092731829574

SVM

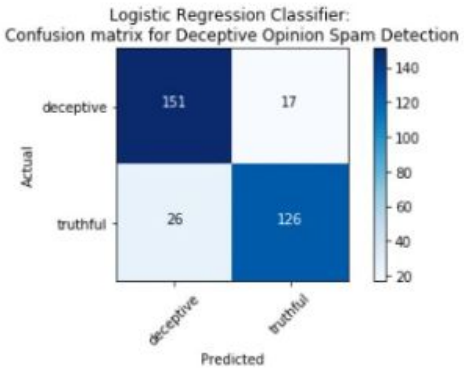
A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. In the project, "Authenticity" as the label data with two categories Deceptive and Truthful, and Reviews in the training dataset are used to train SVM model to predict the review either deceptive or truthful. The project evaluates the linear SVM with Countvectorizer and TF-IDF vectorizer, by comparing macro-average F1-score, linear SVM with TF-IDF vectorizer has better performance with macro-average F1-score 0.90.



	precision	recall	f1-score	support
deceptive	0.87	0.92	0.90	151
truthful	0.93	0.88	0.90	169
accuracy			0.90	320
macro avg	0.90	0.90	0.90	320
weighted avg	0.90	0.90	0.90	320

Logistic Regression

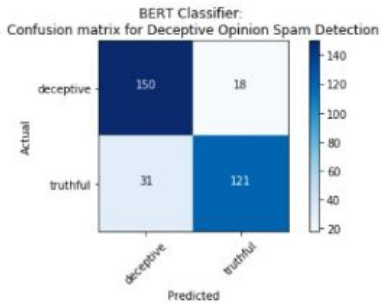
We used another supervised learning model logistic regression to classify authenticity. For logistic regression model, we take the processed textual data and transform to TF-IDF vectors as the input data. The model achieved an accuracy of 86.6%, precision of 86.7%, recall of 86.4%, and f1 score of 86.4%. The confusion matrix and classification report is shown below.



	precision	recall	f1-score	support
deceptive	0.85	0.90	0.88	168
truthful	0.88	0.83	0.85	152
accuracy			0.87	320
macro avg	0.87	0.86	0.86	320
weighted avg	0.87	0.87	0.87	320

BERT

We also used BERT which is a transformer-based machine learning model for our classification problem. For BERT model, it is different from logistic regression. We pass in the textual data and BERT will tokenize the texts by inserting some tokens to indicate the start and end of the sentence. BERT will also use masks to separate different sentences. Each token will be mapped to a token id. Position embeddings are also added to indicate the position of each token. After tokenizing the input, the token ids as well as the attention masks will be fed into the classifier. The model achieved an accuracy of 86.8%, precision of 86.8%, recall of 86.9%, and f1 score of 86.9%. The confusion matrix and classification report is shown below.



	precision	recall	f1-score	support
deceptive	0.83	0.89	0.86	168
truthful	0.87	0.80	0.83	152
accuracy			0.85	320
macro avg	0.85	0.84	0.85	320
weighted avg	0.85	0.85	0.85	320

Model Comparison

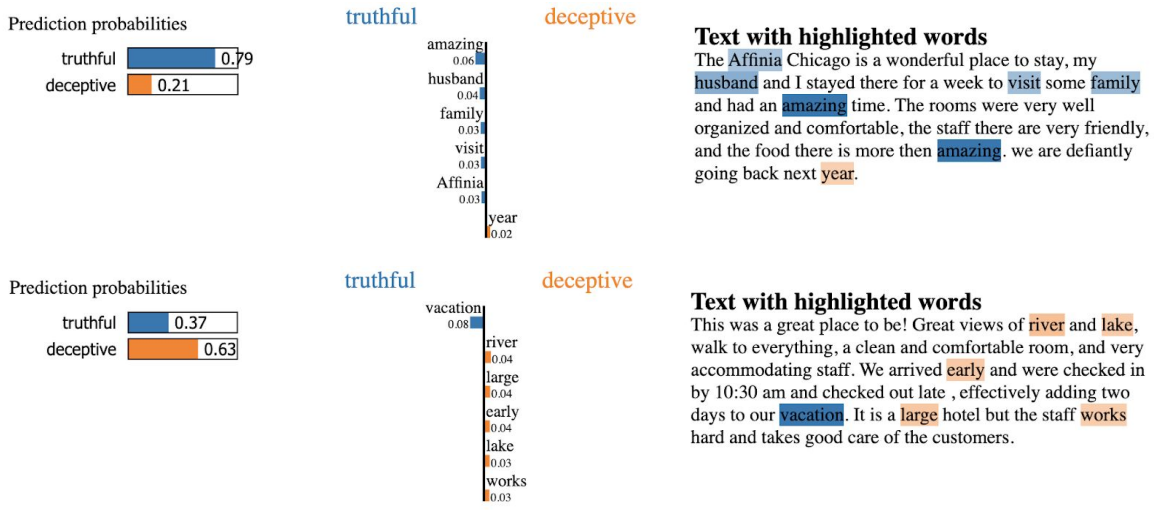
	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.82	0.85	0.82	0.81
SVM	0.90	0.90	0.90	0.90
Logistic Regression	0.866	0.867	0.864	0.864
BERT	0.868	0.868	0.869	0.869

Putting the accuracy, precision, recall and F1 for all models together, we observe that SVM achieves the highest metrics in all four categories, although all models have relatively decent results of over 80%.

Model Explainability

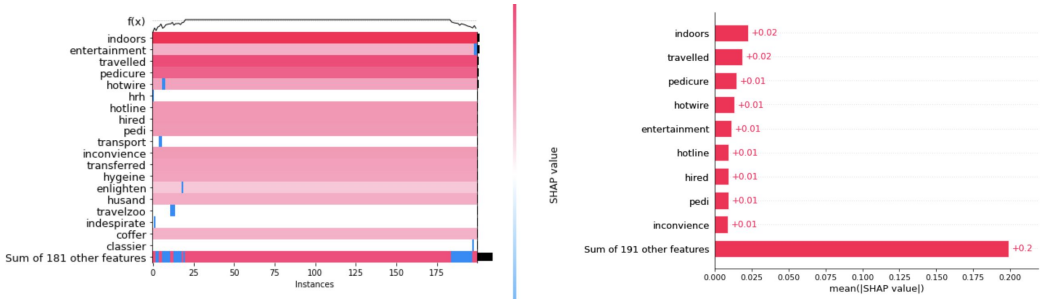
To further understand the model predictions, we applied LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (Shapley Additive exPlanations) explainers.

LIME creates multiple "perturbed" instances around the instance to be explained, and builds a linear model for local interpretation. In the first two plots, we are looking at two reviews, classified as "truthful" and "deceptive" respectively. LIME highlighted the keywords in each review with orange and blue colors to indicate whether these keywords increased the probability for the review to be classified as truthful or deceptive. For example, in the first review, we see that "husband", "amazing", "family" and "visit" led the model to believe that the review is likely to be truthful, whereas in the second review, "river", "lake", "early", "large" and "works" led the model believe that the review is deceptive.



SHAP utilizes Shapley game theory and builds a global explainer with local smoothing. The plots

included below are shapley heatmap and shapley barchart. The vertical axis on the heatmap shows the keywords or features in the text processing context, and the horizontal axis shows the top 200 instances or reviews in the training dataset. The heatmap shows us for each individual instance, whether the keywords increased the probability for that instance to be predicted as truthful (pink) or deceptive (blue). For example, we can see that the word “indoors” increases probability for almost all instances to be predicted as truthful, whereas other words like “hotwire” increases the probability for most instances but decreases the probability for a few reviews to be predicted as truthful. The barchart shows the feature contribution for truthful probabilities across all instances.



Conclusions

Based on the evaluation on the test set, we can observe that BERT model is slightly better than logistic regression and Naive Bayes, but not as good as SVM. This could be because that the dataset is not a complex task. For more complex tasks, BERT model can be expected to perform better. But overall, all the models we trained made pretty accurate predictions on this dataset.

In the future, we could improve by incorporating word2vec embeddings instead of using TF-IDF and count vectorizers, or include more feature engineering techniques before modeling.

Reference:

Pranckevičius, Tomas, and Virginijus Marcinkevičius. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification." Baltic Journal of Modern Computing 5.2 : 221.

Appendix A:

Group Member Responsibilities:
Data Cleaning : Yongzheng Chen
EDA Training Data Set: Zhiyu Lin
EDA Test Data Set: Yiran Liu
Logistic Regression Classifier+Results+Confusion Matrix: Audrey Yuan
Bert Classifier+Results+Confusion Matrix: Audrey Yuan
Naive Bayes Classifier+Results+Confusion Matrix: Yongzheng Chen
SVM Classifier+Results+Confusion Matrix: Yiran Liu
Model Comparison: Zhiyu Lin
Model Explainability: Zhiyu Lin