Airbnb Reviews - Natural Language Processing Project Report

Hai Nguyen
Department of Computer Science
University of Chicago
ndhai@uchicago.edu

Rodrigo Valdes Ortiz
Social Science Division
University of Chicago
rodrigovaldes@uchicago.edu

Abstract

Customer reviews play an important role in many businesses as they reflect the quality of services and helps providers improve their services. However, reviews are typically written in plain text and it is difficult for computers to automatically extract information from them. One standard metric to evaluate reviews is review score, which reflects the sentiment of the review. In this paper, we propose a method that predicts the review score for a spot in AirBnb given its text reviews. The method exploits some unique characteristics of the AirBnb dataset to transform reviews in plain text into a mathematical form and use them to train a machine learning model for predicting review scores. The experiment results show that our methods could achieve up to 69% of accuracy, which is fairly good compared with state-of-the-art works on customer reviews.

1 Introduction

In the last decade, Airbnb has allowed small entrepreneurs to share their houses and personal belonging to strangers. Consecutively, Airbnb generated a market that did not exist in the past, converting regular people into tiny hotel businessmen. One of the main characteristics of its business model relies on user's reviews, which provide an external assessment of the spot.

In this paper, we aim to generate a model to predict the aggregate score of a spot given its text reviews. One of the characteristics of this market is that the guest write a review of the host, and the host makes a review of the guest. Also, considering that Airbnb works similar to a social network, most of the reviews tend to be quite high because they want o to be perceived positively in the network. This creates an interesting problem for natural language processing because the algorithm needs to discern between good reviews to superb reviews. In the same vein, the dataset for reviews is quite unbalanced, making the scarce low reviews very difficult to predict. In this context, we propose some approaches using neural networks as our main tool to predict the aggregate review score of an Airbnb spot.

2 Related Work

Regarding Airbnb reviews, (Zervas et al., 2015) analyzed reviews of 600,000 Airbnb properties. They conclude that most of the reviews are highly positive, above 4.5 stars. In the same vein, for properties cross-listed in Trip Advisor and Airbnb, they found that even when the average score is similar, more properties tend to receive a top rating in Airbnb. In the same vein, (Fradkin et al., 2018) ran experiments to understand strategies to improve data collection process in reviews. They found that one of the main issues with Airbnb reviews is that people who had better experiences tend to write more reviews against people who do not. Then, negative experiences are underrepresented.

On classification of reviews, (Bakliwal et al., 2011) trained a model for movie reviews classification. They got an accuracy of 78.32 %, in the classification of positive and negative reviews utilizing NGrams and POS-Tagged NGram. In a similar framework, (Panichella et al., 2015) developed a scheme for classification of Google Play reviews into useful categories for software maintenance. The merged natural language processing, text analysis, and sentiment analysis. The got precision of 75% and recall of 74%.

With respect to general sentiment analysis, as

summarized by (Bakliwal et al., 2011), researchers have used three main techniques for sentiment classification. First, (Pang et al., 2002) use a syntactic approach utilizing Ngrams. They found that machine learning techniques outperform human-produced baselines when classifying reviews into positive and negative. The maximum accuracy they got is 82.9%. Second, (Turney, 2002) used a semantic approach utilizing part of speech information. In addition, (Benamara et al., 2007) claim that adjectives and adverbs are better for binary classification against utilizing only adjectives. In particular, they use an axiomatic treatment of adverb-adjective combinations based on a linguistic classification of adverbs. They conclude that their approach produces higher accuracy against existing sentiment analysis algorithms (2007). Third, researchers use techniques to extract sentiment expressions and then perform binary polarity classification. Examples of this approach are (Nasukawa and Yi, 2003) and (Bloom et al., 2007).

3 Methods

The key challenge of predicting the review score is transforming the review in plain text into a representation that we can efficiently use it to construct score prediction algorithm. Our intuition tells us that there is a strong relationship between the words used by the reviewers and the review sentiment. That is, if reviewers make use of some words such as "beautiful", "clean" many times then we are very confident to predict that the review score should be very positive. However, one can argue that, sometimes, good words does not implies positive sentiments. For example, reviewers may add negative terms like "no" or "not" to inverse the meaning of very positive words after them. This counter toward our assumption. Fortunately, it is not the case for AirBnb reviews. In fact, the distribution of review scores written by AirBnb users are very skewed. As being shown in Figure 1, a majority of review score falls with the range from 7 to 10. This suggests that most of the reviews are positive so there content should be positive. This reduces the risk of misinterpreting bad comment as a good effect on the review score. This observation let us believe that by looking at some certain words of the reviews, we can predict the score that the tourist will give to the host.

We define *characteristic vector* as a quantity

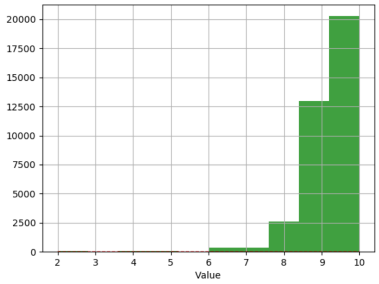

Figure 1: The frequency of review scores

representation of reviews. One can think that this quantity is similar to word embedding. Based on our observation, each entry of the vector is the frequency of a certain word in the review. We use this vector as an input for predicting review scores. Figure 2 shows our approach in more detail. In particular, the approach consists of two key steps.

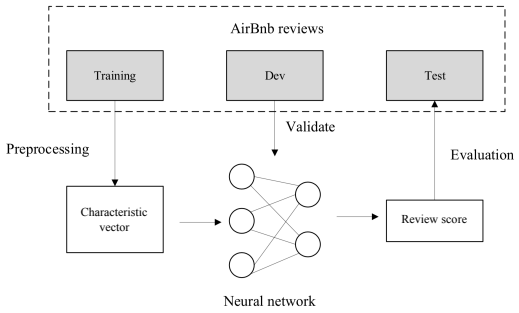

Figure 2: Characteristic vector construction process

- *Preprocessing* AirBnb reviews are decomposed into three datasets, named training, dev, and test set where training are used for constructing characteristic vector, dev test are for training the machine learning model and test set is used for evaluation. During the process, reviews are mapped to spots. The text within reviews are filtered for downstream tasks: words are decapitalized, stop words such as "the", "a" and punctuations are eliminated. Stemming is also applied to reduce vocabulary size and let downstream tasks focus on the sense of the words rather than their deeper meaning.
  Also in this step, we construct a person-word matrix M of size $|V| \times |P|$ where V is the vocabulary of words appear in all filtered reviews and P is the set of spots. Each cell M_{ij} of the matrix is the number of appearance of the word i in the text related to spot

j. Therefore, the column j of the matrix is the characteristics vector of $j \in P$.

- *Training* After obtaining the characteristic vectors from preprocessing step, our next task is to predict the review scores from those vectors using feed-forward neural network. The input of the network are the characteristic vectors and the output are the probability of review score. Our prediction would be the one with highest probability. We use early-stopping technique to improve the training results, that is, we periodically compute the accuracy of the model on dev set and save the model with the best accuracy. At the end, this model is run on the test set for evaluation.

4 Experimental Setup

We use data about Airbnb spots in NY. The data is available on the site `insideairbnb.com`. We have access to two main sources of information. On the one hand, the description of Airbnb spots according to the host. That is to say, a textual description of the spot, such as "quiet, cool, quirky, 1 bedroom apt;" and a summary of the experience that the guest can expect, for example, "enjoy organic breakfast a site located in the heart of NYC and its cultural attractions". Also, it can include the expected interaction with the host, for instance, "I like to be friendly with my guests, sometimes we go together to the museum." There are 90K listings in the data. On the second hand, we have text of all reviews that each spot has received. These reviews can describe the listings by itself, facilities of the building, unexpected characteristics of the unit, or just thank you messages. The number of reviews is 1.5M. Finally, we have an aggregate rating of each spot in the range of 1 to 10, which is the average of its consumer reviews.

For this analysis, we utilize all spots whose sum of reviews have at least 1800 characters. On average, this means that we use spots with at least ten reviews. We define our vocabulary as the most frequent 3000 words in our data. To characterize one spot, we use three different data sources: (1) data including description and reviews, (2) only description, and (3) only reviews. A comparison among them is provided in the next section. We split our data into three sets: a training set, DEV set, and DEVTEST. We preserve 70% of the data for the first dataset and 15% for each of the other two. We utilize Pytorch to generate our neural networks. In our base case, the vector size is 1000, which corresponds to the most important 1000 words, or 80% of the words in the text. For the neutal network architecture, we use one hidden layer of size 512 and use retifier (ReLU) as activation function. The loss function is cross entropy and we use SGD optimizer with learning rate of 0.0001. In addition, we run each model 200 epochs. The source code of our work can be found at `https://github.com/rodrigovaldes/nlp_airbnb`

Text used for vector construction	Accuracy
Review only	68.9791%
Review + Description	65.9497%
Description only	58.1642%

Table 1: Accuracy on testset of model using different source of text

5 Results and Analysis

Table 1 shows the accuracy on the DEVTEST set after training the model with characteristics constructed from different sources: (1) Reviews, (2) Review + host description and (3) Host description. Surprisingly, using the vector of the mix of review and host's description reduce the accuracy by 3% comparing with the model making use of the text from reviews only. We believe that host's description does not relate must to review score. The accuracy of model using only host's description has quite low accuracy. Thus, including them in the characteristic vector would be essentially adding more noise and make the model perform worse.

After the training of the baseline model, we realize that the model predicts several tens (10) that are nines (9) in the data. Also, the distribution of the size of the reviews was different among the tens and the nines. In order to use these insights to improve our model, we develop a new feature according to the number of characters in the sum of the reviews. This new feature is a 500-length vector that depends on the size of the string. To create this vector, we use two steps. First, we generate a random vector of length 500, which is the same during all the model. Second, we multiply the string length of the reviews of each spot by this vector, and the result is appended at the beginning of our vector build from word counts.

However, when training the model, we realized

Model	Accuracy	Time (minute)
Original	68.98%	24
Dimensional reduction	64.67%	10
Review length included	69.28%	25

Table 2: Accuracy and training time of different models

that the training time is pretty long for big datasets so we need make some modification to improve the speed without significantly affecting the accuracy. According to Figure 3, about 70% of the information about word counting is in the most 300 common words. However, there is information that might be important in the words that are not so frequent, but using the 1000-length vector for the training is computationally intensive.

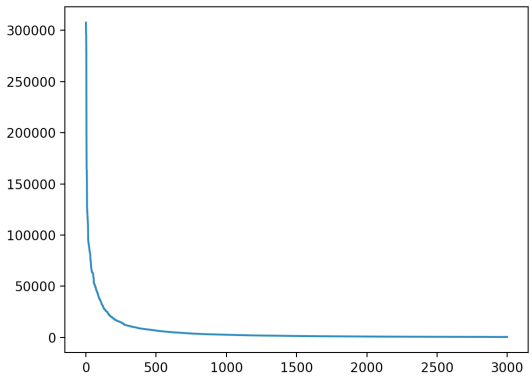

Figure 3: Word frequency

In order to speed up the training, we create a method of dimension reduction that decrease the size of our 1000-length vector. This method is multiplying our vector to a matrix that allows creating local averages. For example, the first 5 entries of a vector become a unique number that is the average of the first 5 entries, and the second element of the new vector is the average of the entries 6 to 10. This process reduces the training time significantly although it has an accuracy cost.

Table 2 shows that adding a 500-length vector which depends on the length of the reviews increase the accuracy 0.3% in the DEVTEST set, which tells that our intuition was correct and the size of the string is related to the score. Also, note that dimension reduction requires 40% of the time of the training time of the baseline model. However, dimension reduction reduces accuracy level by about 4.2%.

6 Conclusion and Future Work

In this paper, we propose a method for predicting the review score of a spot in AirBnb given the plain text of its reviews and the host's own description. The method consists of two steps: we first construct a characteristics vector for representing the text in a mathematical form by counting the occurrence of words. Then, we feed to vectors to a multi-layer neural network to learn the relationship between the text and the score and use it to predict the review score. The experiment results show that our model could reach up to 69% of accuracy which is quite good compare to state-of-the-art models. We also able to speed up the training time by 2.4x with the trade-off of 4% accuracy off.

Even our model achieve a fairly good performance, there is still a gap between it and state-of-the-art models. We believe we should looking for other features apart from the review text to improve the model. However, as we showed, blindingly adding more features will not always improve the accuracy. Thus, more works on figure out the relationship between the score and new features should be considered. Running time is another issues that we should concern. Although dimension reduction works very good, we think other techniques, such as changing optimizer, use POS tagging for adjectives and verbs, and using accelerators are also worth considered.

References

Akshat Bakliwal, Piyush Arora, Ankit Patil, and Vasudeva Varma. 2011. Towards enhanced opinion classification using nlp techniques. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 101–107.

Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato Recupero, and Venkatramana S Subrahmanian. 2007. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *ICWSM*. Citeseer.

Kenneth Bloom, Navendu Garg, and Shlomo Argamon. 2007. Extracting appraisal expressions. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 308–315.

Andrey Fradkin, Elena Grewal, and David Holtz. 2018. The determinants of online review informativeness: Evidence from field experiments on airbnb. Technical report, Working Paper.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall. 2015. How can i improve my app? classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 281–290.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Georgios Zervas, Davide Proserpio, and John Byers. 2015. A first look at online reputation on airbnb, where every stay is above average.