Using MDPs to Generate Airbnb Meta-Reviews

Aaron Effron, Keshav Santhanam, Sophia Chen

Introduction and Motivation

In just the past 10 years, Airbnb has normalized the idea of paying to stay in a complete stranger’s home. While the host is free to advertise her properties however she likes, she cannot control or filter the reviews that are left by guests. Thus, guest reviews are a reliable way to gauge the quality, appeal, and safety of a property.

Often times, it can be time consuming to scroll through the entire set of reviews for a single property. Furthermore, there often is significant redundancy across reviews. Our goal is to generate “meta-reviews” of properties listed on Airbnb using real reviews of Boston and Seattle properties. These meta-reviews will ideally summarize and encapsulate the general consensus and information contained in the set of reviews for a given property.

Problem Definition

We aim to train and test an MDP (Markov-Decision-Process) that will output meta-reviews that appropriately summarize the sentiment and content of the set of reviews belonging to an Airbnb property. In order to do this, we will first build a CFG (context-free-grammar) using parsing rules, and populate terminal symbols using two datasets of Airbnb reviews from Kaggle [1][2]. An MDP state will be a subset of a created review, an action will be the appending of a word phrase to a review (in a way that obeys the CFG), state transition probabilities will be the probability of that word being chosen to be added, and rewards will be based on a measure of fluency of the current phrase and correlation to similar reviews. The MDP will be unaware of transition probabilities and rewards, and thus will learn its policy through Reinforcement Learning. Our inputs in this case would be the significant and frequently found phrases from our review correlation calculation (explained below), and our output would be a meta-review.

Methodology

We will use the pointwise mutual information (PMI) technique to identify the most significant words in each review. This process allows us to find the words that are shared between two reviews, as well as words that appear less frequently (indicating more significance). We will use the Natural Language Toolkit’s (NLTK) implementation of PMI to quickly find the most relevant words for our analysis. Our inputs are two Airbnb reviews, and our outputs are the words or phrases that are most significant and shared between the two reviews.

We will also calculate a correlation between two reviews. This will be a twofold process: first, we compute a score based on the sentiment analysis of each review using NLTK’s Vader Sentiment Analysis tool. Then, after using PMI to find the most relevant words, we calculate synset distance between the relevant word sets. We combine these two scores to calculate a correlation score.

We plan to create and test various reward structures, and evaluate the reviews produced by each alongside those produced by the baseline and oracle.

Evaluation, Baseline, and Oracle

We intend to evaluate our results by having humans read the generated reviews and answer three follow-up questions:

1. Was this review helpful?
2. How likely would you be to stay here?
3. How well did this review encapsulate the most important aspects of the reviews?

The human graders will answer each of these questions on a scale of 1-10, with 1 being the most negative response and 10 being the most positive. Though this method may not scale, we feel it is the most natural way to accurately gauge the effectiveness of our algorithm.

To establish a lower bound on our success, we used TF-IDF (term frequency–inverse document frequency)to find significant words in each set of reviews, then selected random sentences from reviews that included those words to generate a meta-review. As an example, we tested this method using a listing that included the following review, among others:

|  |
| --- |
| I stayed at Casey's apartment for 19 wonderful days together with a business partner. The apartment is spacious, the beds we're very comfortable, the bathrooms, kitchen, living room, air condition, washer & dryer, the view - were just perfect. The building is very comfortable and safe, 10 minutes walk to the red line subway, many restaurants and cafes around, gym and lounge in the building. Casey himself was always available online, was helpful with anything we needed from apartment questions to things to do in the weekend. Overall - it was a perfect Airbnb experience, I highly recommend this listing to anyone looking for a spacious, safe, comfortable place in Boston. |

Though this listing had several similarly detailed reviews, our baseline approach generated:

Fantastic space, would highly recommend staying here if you're visiting Boston.

According to our human graders, this baseline review averaged a 2, 5, and 3 on our evaluation metric. However, a human (our oracle) was able to write a meta-review of this property using the same reviews that were accessible to our baseline algorithm:

This apartment is on the 17th floor in the Innovation district and has amazing views of Boston and the harbor. The red line subway was only a 10 minute walk away, as were several restaurants and cafes. The apartment was spacious and safe. Inside the building there is a gym and lounge area.

This review averaged a 9, 8, and 9 on our evaluation metric. This confirmed our expectation that humans would be able to write far superior meta-reviews to our simple baseline approach. Our model ideally should generate sentences that respect semantic meaning and incorporate significant information from input reviews, coming as close to human performance as possible.

Related Work

Though there has been no meta-review formulation from this dataset, Kaggle users have performed various analyses. For example, one user performed a geospatial analysis of listings, while another user mapped correlations between words and neighborhoods/prices [3][4].

Previous work has been done on sentiment analysis. Cambria et al. describe the many different methods that have been explored with text analysis, including using SVM’s to classify positive or negative movie reviews (Pang et al.) and knowledge bases such as WordNet to perform bootstrapping [5]. Other methods like keyword spotting and lexical affinity are also mentioned [5]. Our technique uses keyword spotting and synset distance between keywords in different reviews.

Lastly, Dong et al. built a neural network model to generate user reviews given a set of user attributes and a corpus of Yelp and Amazon review data [6]. This work is relevant in considering their methodology in extracting the most important information from reviews.