Optimizing Airbnb Listing Title Generation based on Multifeature Selection

Charles Pan, Khalil Miri, Adi Sidapara

CS 221 - Artificial Intelligence: Principles and Techniques

Problem Statement

Airbnb is an online marketplace for listing homes, apartments, and other properties for stays, is quickly becoming one of the most popular options for vacation and short-term lodging.

Hosts can specify and guests can narrow down listings by type of lodging, dates, number of rooms, price, etc.

On Kaggle, a user has compiled all New York City Airbnb listings in 2019 including data such as:

- Name of the listing
- Location of the listing
- Price
- Average number of reviews

Using data we hope to:

- Predict the average number of reviews based on the listing name
- Change words in the listing name in order to increase the expected average number of reviews for the host

https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

Approach and Challenges

Cleaning The Data

Cleaning the Titles:

The dataset came with the titles for every listing, but these user generated titles varying significantly in their wording and terminology. To clean the data for proper analysis and usage, we:

- Turned all uppercase characters into lowercase characters
- Split words connected by punctuation (3-room-apartment -> 3 room apartment)
- Converted shorthand spelling to relevant words (w\ -> with)

Disparity in Locations and Reviews:

The Airbnb dataset included all listings from all five boroughs of New York City, which proved to have lots of variation across all variables. To make our task more feasible, we split the dataset by:

- Region: created a separate dataset for each borough of New York City
- Number of Reviews/Month: split each borough's dataset into two equally sized datasets corresponding to number of reviews/month

Continuous to Discrete Outputs

Our dataset's outputs were the average number of reviews per month, which was a non-integer typically between 1-10. To evaluate our performance, we changed our evaluation metric to a categorical one by:

- Creating five "bins" for the average number of reviews per month (corresponding to very low, low, average, high, very high)
- Assigning every listing to one of the bins such that all the bins have equal size

First Approach: Linear Regression

Our first approach was to run linear regression on our dataset. We:

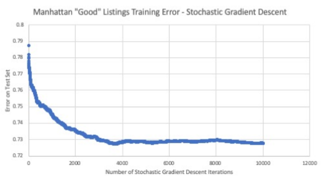
- Created a dictionary of all words found in listing titles of our training set
- Ran stochastic gradient descent over the training set 10000 times
- Evaluated our linear regression model over test set

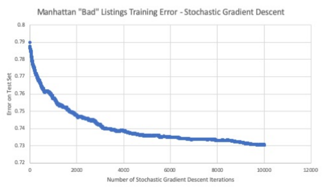
Second Approach: LSTM

This approach was planned as follows:

- The LSTM model is trained on titles and their reviews per month.
- The titles are encoded by a GloVe model using spaCy trained on a global corpus of Wikipedia articles.
- The backtracking search substitutes keywords with their top 10 synonyms by highest cosine similarity
- Each modified title is evaluated with the LSTM model and the optimal synonym substitution is returned

Results





These two graphs show the training error as stochastic gradient descent (SGD) was being run on linear regression for "good" and "bad" performing listings in Manhattan. We found:

- The good listings converged faster towards its final training error than the bad listings, indicating their titles likely had similarities than the bad listings.
- Both the good listings and bad listings converged to about 73% error, which is just slightly better than random chance (80%), indicating a relatively faulty model.

Linear Regression Accuracy

Borough	Accuracy (High, Low)
Brooklyn	26.769%, 24.445%
Bronx	33.333%, 32.622%
Manhattan	25.975%, 24.343%
Queens	33.081%, 26.605%
Staten Island	38.462%, 39.316%

LSTM Accuracy

Borough	Accuracy (High, Low)
Brooklyn	24.574%, 23.479%
Bronx	15.909%, 27.272%
Manhattan	21.514%, 22.744%
Queens	27.511%, 22.271%
Staten Island	12.5%, 12.5%

The above charts show the accuracy for each dataset for both linear regression and LSTM. As shown, the linear regression performed significantly better than the LSTM, indicating a more linear correlation between title's words and its performance. We also found that the less samples there were in a dataset, the better the linear regression/LSTM models performed.

Example of Correctly Predicted Title:
"spacious stylish top floor 1br whole apt"
Predicted: 2.186, Actual: 2.16

Example of Incorrectly Predicted Title:
"charming bedroom in harlem"
Predicted: 2.940, Actual: 6.19