Quantitative Analysis of Hotel Reviews

Sam Hughes
Northeastern University
hughes.sa@northeastern.edu

Wang Xie
Northeastern University
xie.wa@northeastern.edu

Pengkun Ma
Northeastern University
ma.peng@northeastern.edu

Ryan Quinn
Northeastern University
quinn.rya@northeastern.edu

Abstract

Using 20,491 hotel reviews and their corresponding rating from Trip Advisor, we utilized machine learning and natural language processing to not only estimate the review rating, but also to quantify the key features to a beneficial and detrimental hotel experience. We employed a transformer model with the raw review text and an elastic net model with input of hotel traits and amenities in addition to the sentiment. We found that the transformer model accurately predicted review ratings, but struggled to quantify key factors, while the elastic net model offered less accurate predictions, but quantifiable hotel traits and amenities. Using the elastic net model, we observed that location, employees, and dining options are crucial to a positive hotel experience.

Introduction

Motivation

Hotels aim to satisfy their guests by catering to their needs. Even still, it is not always clear which factors are most essential to customers. The hotel traits and amenities that customers care most about are apparent in their written reviews, where they detail what they liked or disliked about a hotel. By using machine learning and natural language processing, we can accurately estimate the score of a new review. However, the estimation of a review alone leaves minimal benefit, as a customer typically provides their score along with the written review. Instead, we use the machine learning model as part of statistical inference, quantifying the value of numerous hotel traits and amenities commonly mentioned in hotel reviews.

Hypothesis

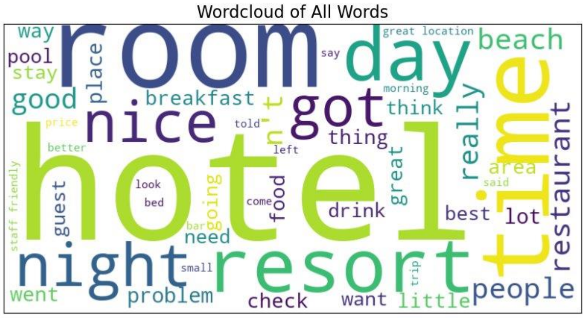
Our goals with the analysis are two-fold. First, we hypothesize that we can accurately estimate the rating of a hotel given a review. Second, by using the model, we believe that we can reverse engineer the results of our machine learning models to quantify the traits that are most beneficial and detrimental to a hotel experience, offering valuable feedback to hotels on what matters most to customers.

Data

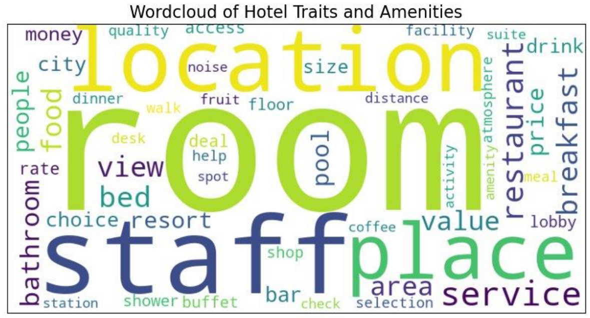
Overview

The dataset contains 20,491 hotel reviews and their corresponding rating from Trip Advisor, available on Kaggle. In the initial dataset, the reviews contain light text preprocessing, notably already in lowercase.

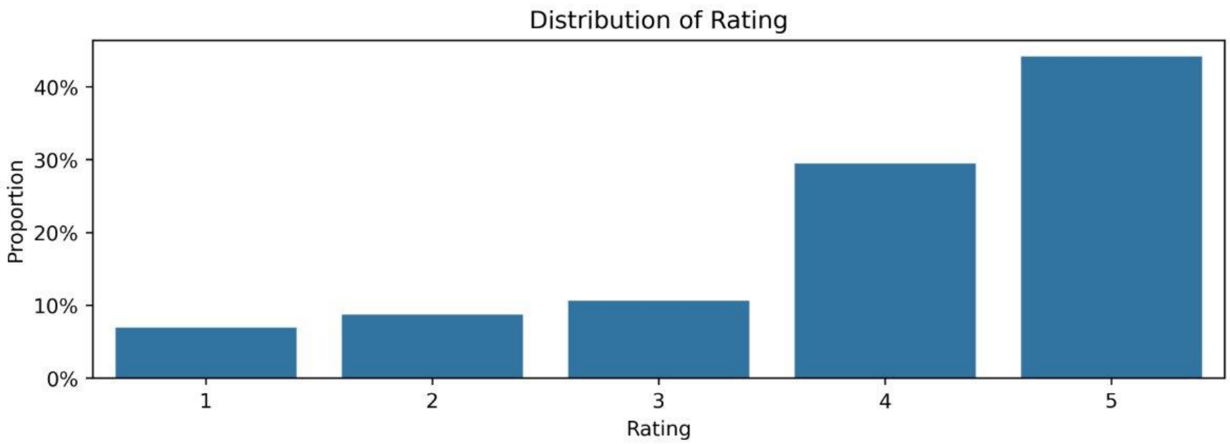
Wordcloud of All Words



Wordcloud of Hotel Traits and Amenities



From the corpus, we see many expected words frequently appearing in the review text, such as ‘hotel’ and ‘room.’ We further preprocessed to extract meaningful hotel traits and amenities from our dataset, where we once again see many expected words, such as ‘location’ and ‘staff’.



The rating column is notably skewed, with higher rating scores more frequently occurring. This was surprising to us, as we intuitively assumed that the phenomenon of sampling bias would lead to a bimodal distribution of frequent low and high star ratings, as those with strong opinions on the hotel would be more likely to leave a review. However, this is not observed in the dataset.

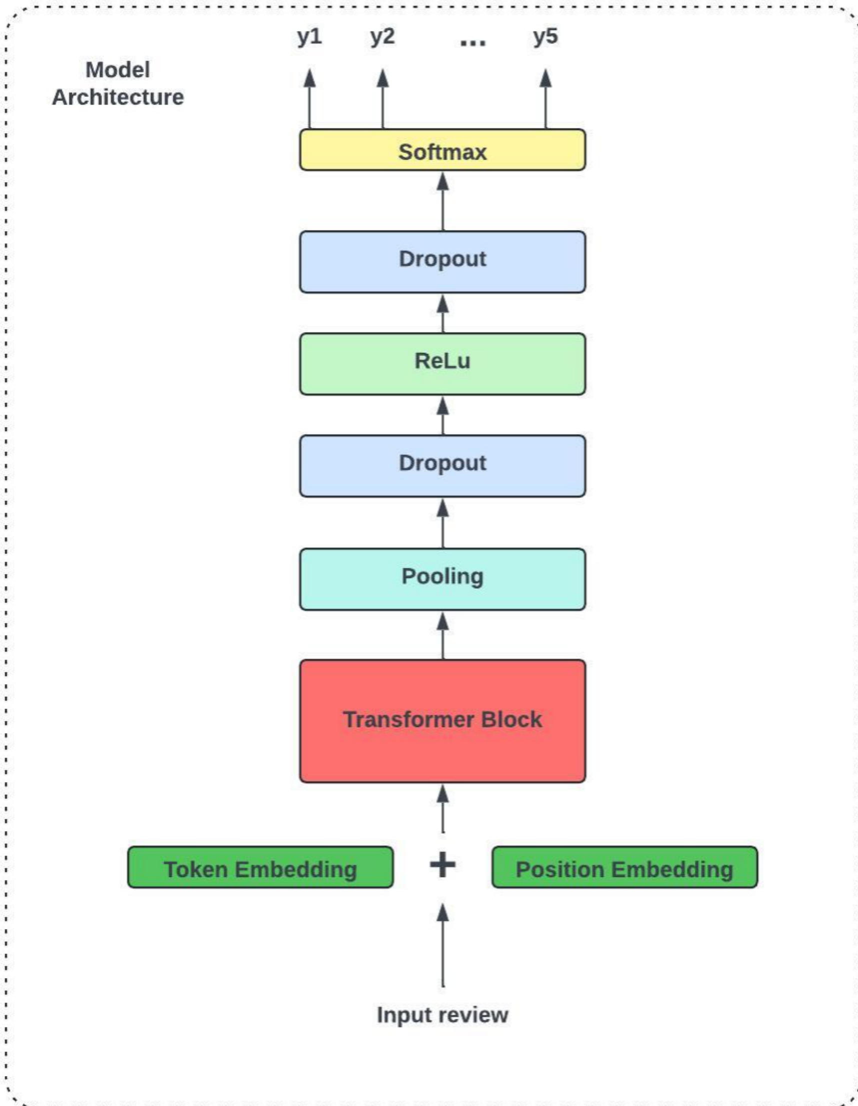
Methodology 1: Transformer Model

Data Preprocessing

In the transformer model, we used the built-in tokenizer from TensorFlow [1] for tokenizing the text reviews and converting them into sequences of tokens. We then padded the sequences into uniform length of 200 tokens. Ninety percent of reviews were 209 tokens or less, and therefore we felt as if a uniform length of 200 tokens was a desirable choice to balance model simplicity without the loss of information.

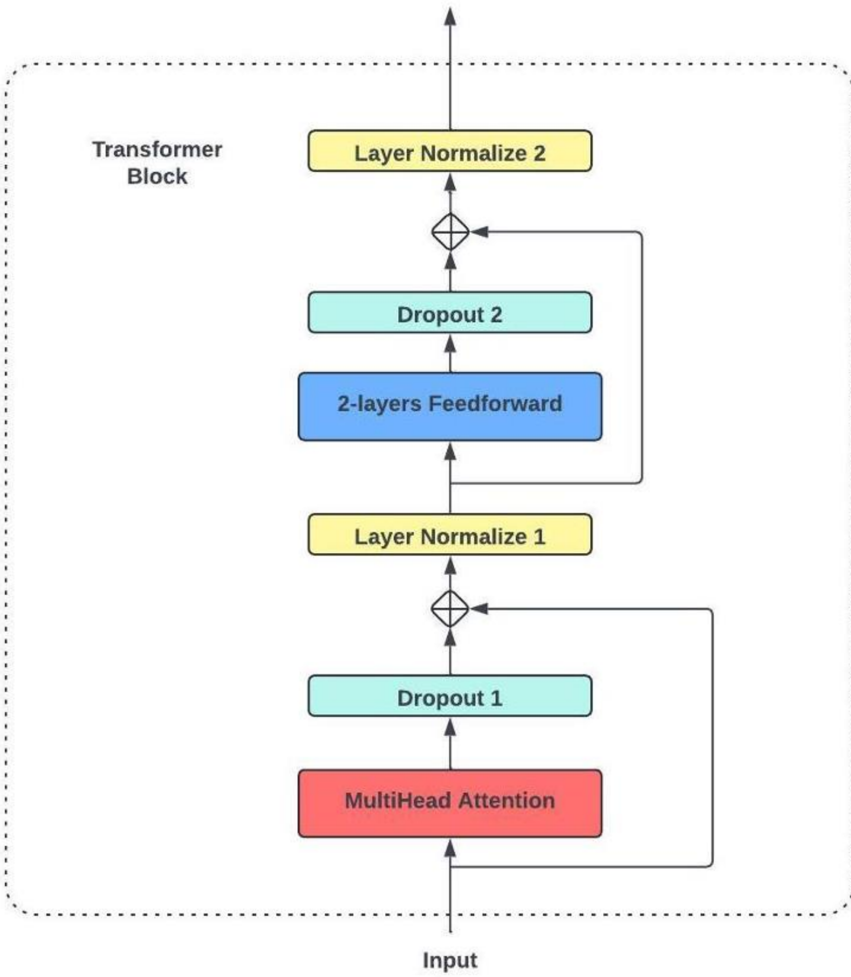
Transformer Model Architecture

We utilized a transformer-based architecture [2] for our language classification model. The preprocessed input review is first sent into an embedding layer. The embedding layer consists of token embedding and position embedding. Each of the embeddings has an input dimension equal to the vocabulary size and an output dimension set to 32. The token embedding and position embedding would be added together to form the total embedding before sending it to the transformer block. The details of the transformer block will be explained in the next section. The output of the transformer block is sent to a global average pooling layer, followed by a dropout layer with a rate of 0.1. After, there is a densely connected neural network layer with 20 neurons and a rectified linear unit activation function. We have another dropout layer with a rate of 0.1 before we reach the last densely connected neural network layer, with five neurons and SoftMax activation function, that produces the final classification output of our model. Each of the five values of the final output is the predicted likelihood of the input review belonging to each class of rating from one to five. The following figure is an overview of the model architecture.



Transformer Block

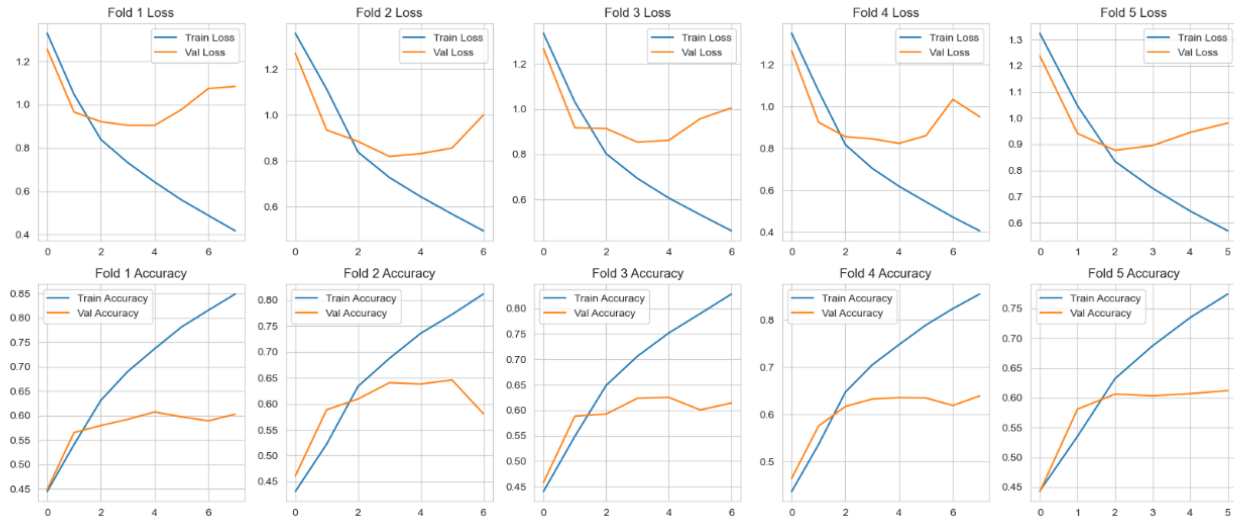
Our transformer block comprises six layers. The input is sent to a multi-head attention layer with two attention heads and key/query dimension of thirty-two, matching the output dimension of our embedding layer as mentioned in the previous section. Next, the output of the multi-head attention layer is sent to a dropout layer with a rate of 0.1. The resulting output is then combined with the original input through a residual connection and passed to a normalization layer. After the normalization layer, we have a two layer fully connected feedforward neural network. The first layer of this network is a densely connected neural network layer with thirty-two neurons, utilizing a rectified linear unit activation function. The second layer, also densely connected, contains thirty-two units but does not apply any activation function. Subsequently, the output of the feedforward network is sent to another dropout layer, and the output is combined with the output of the first normalization layer through a residual connection. Finally, the second normalization layer will produce the final output of the transformer block. The following figure is an overview of the transformer block architecture.



Model Fitting and Evaluation

In exploring model configurations, we conducted an extensive ablation study to examine the impact of various normalization strategies on the model's efficacy. We compared the effects of applying layer normalization before the multi-head attention and feedforward neural networks layers (pre-normalization), after these layers (post-normalization), and only after the feedforward neural network layers. Our findings indicate the post-normalization configuration consistently outperformed the others in terms of convergence and performance matrices.

We incorporated a five-fold-cross-validation, to strengthen the model's robustness against overfitting. Early stopping was employed to mitigate overfitting with a patience parameter set to three epochs. While our models were configured to train for up to ten epochs, they typically ceased training around the fifth epoch due to early stopping triggers. Our evaluation graphs show a trend where training loss decreases and training accuracy increases consistently across all folds, affirming the model's ability to learn effectively. However, we observed an upward trend in validation loss, suggesting that our model is still overfitting and may benefit from further regularization or training on a more diverse dataset.



To provide a more practical interpretation than the log loss evaluation function and to facilitate a better comparison with the elastic net model, we computed both the mean absolute error (MAE) and mean squared error (MSE) of the predicted review rating against the true review score. The MAE was calculated to be 0.49, while the MSE was found to be 0.43.

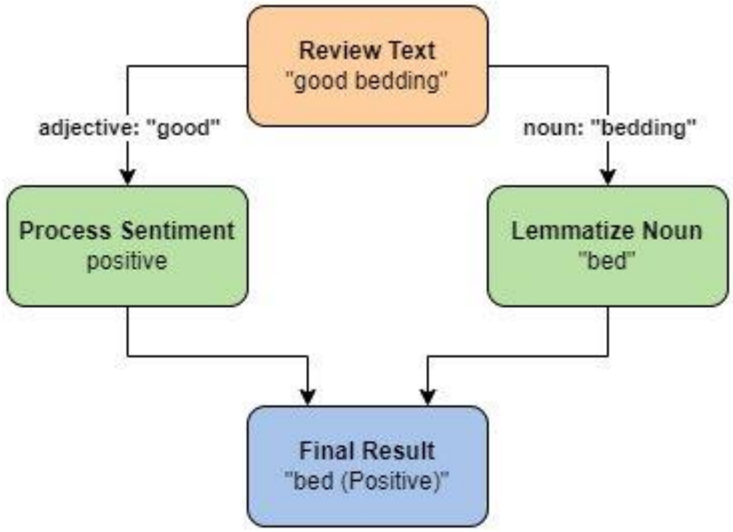
Model Challenges

While the transformer model accurately predicts the hotel review rating, we noticed challenges in using the model to quantify hotel traits. As an example, the review 'good staff, bad bed' is predicted as a rating of 3.05. Compare this to the review of 'bad bed, good staff,' which is predicted as a score of 2.44. From a human perspective, we observe that these reviews say the same thing in a different order, yet the model predicts a 0.61 difference in score. Further, the review 'good bed, bad staff' results in a predicted score of 3.02, just a 0.03 difference from the 'good staff, bad bed' review. From these three reviews, we can see that the transformer model relies on the positioning of the adjectives used in the review as opposed to the content that is referred to positively or negatively. Therefore, we determined that the model does not give reliable quantifications of hotel traits.

Methodology 2: Elastic Net Model

Data Preprocessing

To better understand the impact of hotel traits and amenities, we transitioned from using the transformer model to an elastic net model [3]. In doing so, the input features must be restructured. Instead of the input format of the transformer model, we used a binary bags of words format, where we treated each hotel trait along with its sentiment context mentioned across the corpus as a feature in our model, with a value of 1 if it appeared in the given review, and 0 if it did not appear.



For example, let us take an example review stating that there was 'good bedding.' First, the review is parsed by spaCy's (https://spacy.io/) tokenization process. In doing so, 'bedding' is lemmatized to 'bed,' and 'good' indicates that the sentiment of the text is positive. Thus, the review would have a value of 1 for the 'bed (Positive)' feature. This would be repeated for all nouns in the review text, except those with a neutral sentiment. Further, nouns without a document frequency of at least one percent were removed, and any nouns that did not offer clear hotel improvement feedback (such as 'husband' and 'daughter') were removed. After preprocessing, 65 features remained, all being hotel traits and amenities along with their respective sentiment context. With this new architecture, we could effectively use the coefficient values of the model to quantify the impacts of each hotel amenity and trait mentioned in the reviews.

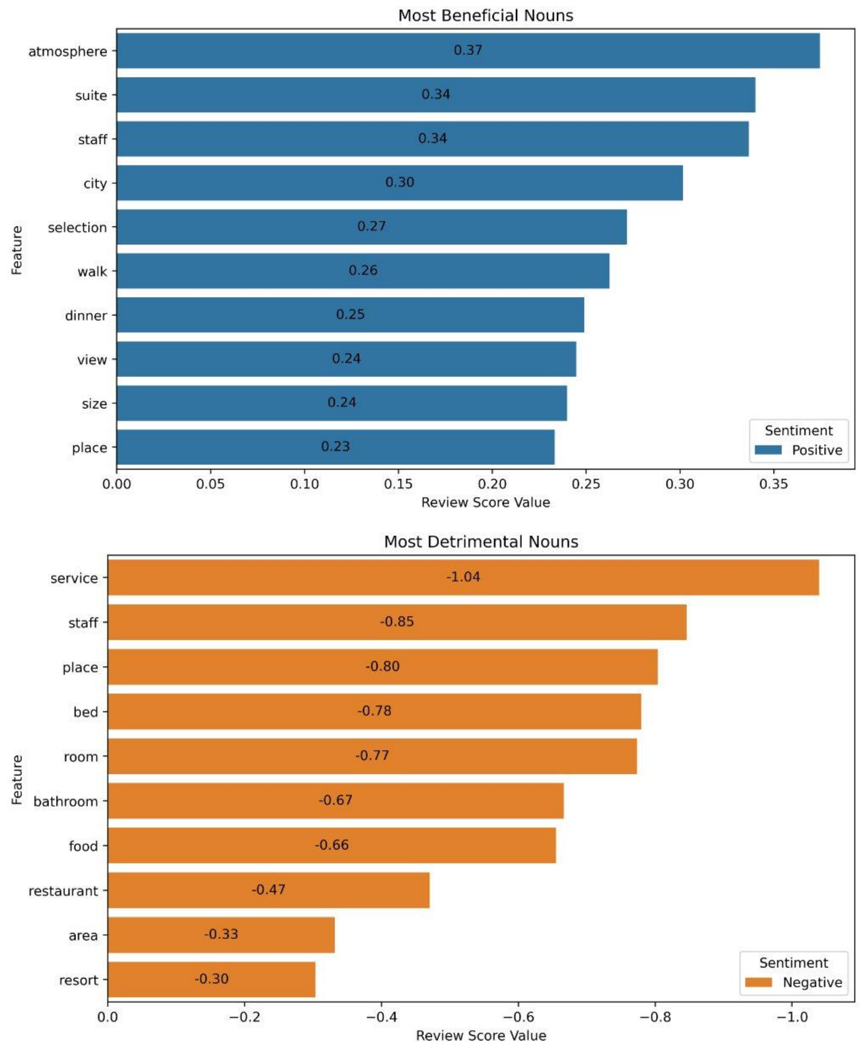
Model Fitting and Evaluation

The elastic net model was tuned to find the best L1 ratio from 0.0 to 1.0 and alpha from 0. hyperparameters using 3-fold cross validation. The hyperparameter search was completed using optuna (https://optuna.org/), which systematically completed 1,000 trials of hyperparameters, and aimed to minimize mean squared error. Under this modeling architecture, we assume the rating target is a continuous variable, which helps quantify the value of hotel traits and amenities. In the end, an L1 ratio of 0 prevailed, meaning that the best model was of ridge regression architecture, with an alpha of 13.16. With this model, we achieved a mean absolute error of 0.90 and a mean squared error of 1.25. While these metrics are worse than the transformer model, it is important to note that only hotel traits were used as input to the model. The goal of this model was not necessarily to achieve the most accurate predictions, but instead to quantify the value of the hotel traits mentioned in the review.

Results

Quantifying Hotel Traits and Amenities

With the elastic net modeling architecture and our binary bag of words input space, we can easily quantify the value of each hotel trait and amenity mentioned in the corpus by observing the coefficient values of the model. The following plots show the hotel traits and amenities mentioned that are most beneficial to the resulting review rating.



According to our model, ‘atmosphere’ is the most beneficial hotel trait when used in a positive sentiment, with a value of 0.37. This means that if a review mentions ‘atmosphere’ positively, it is expected that the rating will increase by 0.37, provided that the rest of the review remains constant. Conversely, ‘service’ is found to be the most detrimental trait when it is negatively mentioned in the hotel review, with an expected decrease of 1.04 stars, provided the rest of the review remains the same. It is notable that many of the most beneficial traits are related to the location of the hotel, such as ‘city,’ ‘walk,’ ‘view,’ and ‘place.’ This suggests the importance of a hotel in a premier area, with many nearby attractions. Likewise, dining is a crucial contributor, with ‘dinner’ denoted as a key beneficial trait while both ‘food’ and ‘restaurant’ appear among the top detrimental nouns. This is a clear indication that hotels should have premier dining options, either on-site or at a nearby location. Most crucially, we see the dual importance of ‘staff,’ which places as a top beneficial and detrimental hotel trait. Therefore, it is highly recommended for hotels to have exceptional employees.

Discussion

Summary of Findings

In this study, we employed a dual-method approach to analyze hotel review ratings. The elastic net model efficiently managed complex data, identifying key traits and amenities, such as cleanliness, staff, and location, that had a substantial impact on the ratings. Despite the elastic net model’s ability to highlight key factors, it did not capture the full complexities of natural language, such as nuances in grammar and sentiment. To address this, we utilized the transformer architecture.

Our transformer model worked to both enhance our ability to predict review scores with greater accuracy and to explore the subtleties of emotional expression. For instance, the word ‘bed’ changed significantly when paired with ‘good’ or ‘bad,’ demonstrating the model’s sensitivity to adjectives. However, the model highlighted a bias towards the sequence of descriptors. For example, switching ‘good staff, bad bed’ to ‘bad bed, good staff’ changed the predicted score from 3.05 to 2.44. Nonetheless, this bias could also be perceived as a pattern that reviewers tend to follow since they tend to prioritize their most significant opinions at the start of their review. This insight into the model’s performance and reviewer behavior offers a potential area for improvement.

Implications of the Analysis

Our study offers valuable insights on how machine learning can enhance services in the hospitality industry. The transformer model we developed accurately predicts review ratings from textual feedback, which can serve as a powerful tool for hotel management. It could also be incorporated into a continuous feedback loop, where reviews are fed into service improvement strategies. On the other hand, the elastic net model gives insight into the traits that customers value the most. For instance, the model finds that a good atmosphere is strongly impactful to a beneficial hotel experience, while bad service is detrimental to the review scores. It would be advisable for a hotel to focus on a lively atmosphere to enhance its service and avoid negative reviews.

Future Ideas

One improvement to the quantification of hotel traits may have been clustering similar words together, which would help quantify traits that are synonymous in the context of hotel improvement. We briefly attempted this on pretrained embeddings but found that the clusters did not make sense from a practical perspective, often containing dissimilar words. In future iterations of the project, we plan to further experiment with word embeddings to cluster similar hotel traits. Another alternate idea would be the use of a transformer architecture in review summarization. While the transformer struggled to quantify hotel traits, we believe that the model may be better suited to summarize a corpus of reviews pertaining to a single hotel.

Supplementary Information

Acknowledgements

This project was completed as part of CS 6120: Natural Language Processing at Northeastern University, instructed by Professor Uzair Ahmad.

Dataset

The dataset used in the project can be found on Kaggle:
https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews

Coding Repository

Our coding repository is available on GitHub: https://github.com/NLP-team-22/Hotel_Review_Project

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. *In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 265-283).

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. *In Advances in Neural Information Processing Systems* (Vol. 30). [No pagination]

[3] Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320. doi:10.1111/j.1467-9868.2005.00503.x