Predicting Yelp Restaurant ‘Ambiance’

Jeremy Yeung , Angela Guan , Amy Jung
University of California, Berkeley

Abstract

To predict restaurant ambiance, we seek to predict particular attributes of restaurants based off of text from Yelp reviews. We use Bag of Words, TF-IDF, and BERT embeddings of review text to predict the value of attributes using Multinomial Naive Bayes, Logistic Regression, BERT, and Multichannel Convolutional Neural Networks. By comparing the accuracy of each model on different attributes, we find Logistic Regression to outperform the more complex models.

1. Introduction

The question that gets asked in every social setting is: “What do you want to eat?” However, the answer is typically not what but where. When looking for restaurants, customers turn to search engines and smartphone applications like Yelp to help them decide. This decision-making process focuses not only on the type of cuisine, but also the type of restaurant — whether the restaurant has a dress code, outdoor seating, is open late, takes reservations, has parking, is good for kids, etc — essentially, the restaurant’s ambiance.

Predicting restaurant ambiance takes into account the steps that are required to search for the perfect restaurant: a) whether the restaurant has the attributes one is looking for, and b) whether the reviews substantiate the restaurant’s listed attributes. A restaurant may have a specific attribute toggled as “True”, but the customer reviews may say otherwise. In this project, we use natural language processing and deep learning to detect attributes of a restaurant from its user reviews.

2. Background

In recent work, BERT has offered significant advances in NLP. BERT, also known as bidirectional encoder representations from transformers, is a transformer-based machine learning framework for natural language processing designed to pre-train deep bidirectional representations from unlabeled text by conditioning on both left and right context in all layers [1]. BERT uses self-attention to create contextualized embeddings for words and sentences. BERT’s use of transformers means it is also parallelizable, allowing it to be trained much quicker — a huge improvement from the previous serializable LSTM and RNN encoder-decoder architectures. Additionally, BERT has been pre-trained on a large amount of data, with BERT-base that has been trained on about 800 million words and BERT-large that has been trained on Wikipedia with about 2.5 billion words. BERT has yielded state-of-the-art results on various natural language processing tasks (a 7.7% point improvement on GLUE score, 4.6% improvement on MultiNLI accuracy, 5.1% point improvement on SQuAD v2.0 Test F1) [1].

A case study [2] performed attribute detection on Google reviews via distant supervision in which they used crowdsourced attribute labels of restaurants as labels for the review text without direct annotation of the reviews. In addition, they used neural methods [3], such as Convolutional Neural Networks (CNN), for distantly supervised relation extraction. They proceeded to use review-level attention to pay attention to reviews related to the attribute of interest, and the attention weights helped explain how relevant reviews are captured by the model. In addition, the attribute classification was related to aspect-based sentiment analysis work [4]. Across the attributes, there contained both objective and subjective attributes — objective attributes such as *hasOutdoorSeating* extract detailed facts from the text, while subjective attributes such as *feelsRomantic* require detecting granular information from review text as well as sentiment analysis to learn overall polarity. We focused our study on labeled attributes given by restaurant owners.

3. Methods

Our goal is to train a model on the restaurants' reviews in order to predict labels for an attribute. The inputs of our model are the reviews grouped and concatenated by restaurant (restaurant-level data), and the outputs of our model are True or False labels for the respective attribute.

Since our model is performing binary prediction (True or False for an attribute), we use accuracy as our main metric for model performance. We evaluate the accuracy, precision, and recall of the predictions of different attributes and compare the models below.

3.1 Dataset details

We use the Yelp review dataset and the Yelp business dataset from (https://www.yelp.com/dataset/download), which contain the customer reviews and restaurant business attributes respectively. Our dataset contains 2,845,251 reviews from 50,314 restaurants in selected cities across North America.

In addition, we subset our data to include restaurants with at least 50 reviews to collect enough text for training the model. We are assuming that if a restaurant has an insufficient amount of reviews, we may not be able to effectively predict attributes based on reviews. While there are over 30 attributes, including numerical and categorical attributes, there is class imbalance and sparsity present in terms of restaurants' labeling for attributes such as *CoatCheck, BYOB, GoodForDancing,* and *BusinessAcceptsBitcoin*. We choose to focus on binary attributes which are labeled in the majority of the restaurants, and those which are indicative of restaurant ambience.

Our filtered dataset focuses on the following attributes: *OutdoorSeating*, *GoodForKids*, *HasTV*, *RestaurantsReservations*, and *RestaurantsGoodForGroups*. Each instance consists of one restaurant, whether an attribute exists, and concatenated reviews for that restaurant. These datasets were made to have an equal number of rows in each group (attribute = True, attribute = False).

3.2 Model Architecture

BERT

Our BERT models use the 'bert-base-uncased', a pre-trained BERT model that has 12 encoder layers, 768 hidden dimensions and 12 self-attention heads. For pre-processing, we use the 'bert-base-uncased' tokenizer to preprocess the restaurant reviews. In addition, we add special tokens and truncate reviews longer than the max length of 512. The maximum input sequence length is applied before feeding into the model [5]. Testing different hyperparameters led to using the Adam optimizer, binary cross entropy loss, learning rate of 0.0005 and a run time of 10 epochs.

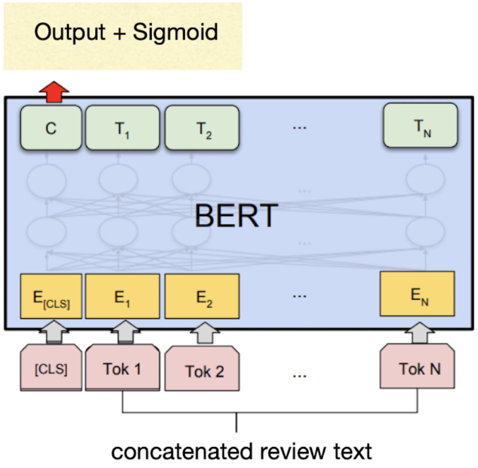

Figure 1: BERT Model

CNN

We use the Keras tokenizer and remove non-alphabetic, punctuation, stop words, and short tokens to preprocess reviews. The CNN architecture consists of three channels that each have an embedding, convolution, dropout, max pooling, and flattened layers that are then concatenated together. A pool size of 2 is used due to compute capacity limitations. We use the Adam optimizer, binary cross entropy loss, learning rate of 0.001 and run time of 10 epochs [9].

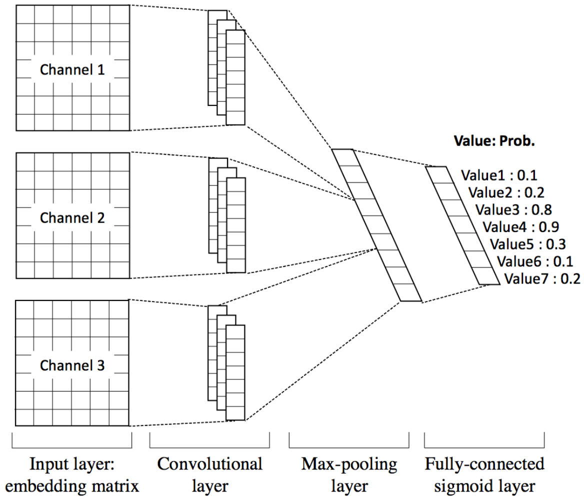

Figure 2: Multichannel CNN Model [8]

3.3 Model Training Details

To establish a baseline, we use Naive Bayes on the vectorized words using Bag-of-Words (BoW). To start, we use RegEx to remove stopwords, symbols, and numbers. This will improve performance and decrease the vectorized text size. Then we apply the NLTK tokenizer to tokenize the text. CountVectorizer is used for BoW to vectorize the reviews to create features, which is split into a 70/30 train-test split before being fed into the Multinomial Naive Bayes model.

After establishing the baseline, the next model we build uses the same preprocessed and BoW-vectorized text which is fed into the Logistic Regression model. We seek to improve this further by using Multinomial Naive Bayes and Logistic Regression on vectorized reviews using Term Frequency-Inverse Document Frequency (TF-IDF). The BoW vectorizer keeps track of frequency of words, while TF-IDF measures frequency of term in a document multiplied by the inverse document frequency (ie. provides the importance of the words by considering the rareness of a term).

We know that the BoW and TF-IDF approaches are not the best way for capturing context which may be important when classifying the Yelp reviews, so we use BERT because it excels at capturing these semantics. Our first BERT model uses the pre-trained ‘bert-base-uncased’ tokenizer and froze all its layers except for a single dense layer to predict our labels. Our second BERT model is identical but unfroze the pre-trained layers to fine-tune on our data. The third BERT model freezes the pre-trained layers but incorporates two more dense layers to learn more on the reviews. The last BERT model keeps these extra dense layers and unfroze the pre-trained layers, making it the biggest model so far and theoretically most versatile.

Since our data consists of multiple reviews per restaurant, we also examine predicting attributes based on review-level data, which would result in larger amounts of samples but shorter text, reducing the context BERT can learn from each review. The motivation for this is due to the max length in the BERT tokenizer could be reached if more than 50 reviews were concatenated, thus losing information per restaurant. However, we find that the review-level BERT model did not perform as well as restaurant-level in our baseline, so we proceed with mainly restaurant-level analysis.

The last type of model we built is a CNN. BERT's CLS token for each review may not capture specific information about the attributes, so finding relationships between certain words using a CNN may improve accuracy. We train our CNN on both the review-level dataset and the business-level dataset. Due to time constraints, only results for the best and worst-performing attributes from the previous models are captured.

4. Results

	Attributes				
Model Types	*Outdoor Seating*	*Good For Kids*	*Has TV*	*RestaurantsReservations*	*RestaurantsGoodForGroups*
NB (BoW)	0.6975	0.8310	0.7154	0.7975	0.7225
NB (TF-IDF)	0.6627	0.7607	0.6992	0.7359	0.5279
LR (BoW)	0.7194	0.8659	0.6918	0.7933	0.7933
LR (TF-IDF)	0.7320	0.8637	0.7346	0.8239	0.7598
BERT	0.5944	0.6559	0.5809	0.6375	0.5854
CNN	0.5692	0.5277	N/A	N/A	N/A

Table 1. Test Accuracies

NB and LR represent Naive Bayes and Logistic Regression, respectively.

Across all the attributes we analyzed, the restaurant reviews detected the attribute *GoodForKids* with the highest accuracy. This could be due to the model learning more context and mentions about kids in the reviews. For example, some reviews include: "Kids options look good too! My kids love pizza. The Sunday brunch is phenomenal and kids eat free. My wife and I were out without kids and really wanted an awesome steak!" In contrast, the attribute *OutdoorSeating* had the lowest accuracy using the Naive Bayes and Logistic Regression models. This could be due to the nature of reviews containing text more focused on food, specific dishes, or customer service. As a result, the BoW and TF-IDF vectorizers learn limited amounts of information from the low frequency of words pertaining to *OutdoorSeating* when compared to *GoodForKids* and *RestaurantsReservations*, which seemed more frequently discussed and led to better detection.

Surprisingly, the BERT models could not surpass the simpler Naive Bayes and Logistic Regression models in accuracy. Out of all the BERT models, the first model with frozen pre-trained layers and the single dense layer performed the best. Compared to the other BERT models, this model had only 769 trainable parameters, compared to 213,377 trainable parameters in the next frozen model, and 109 million trainable parameters for the unfrozen models. We think that the maximum token length may have been exceeded for some restaurants. In addition, the CLS token may not be ideal for our task because it will capture overall sentiment, which is too broad for predicting specific attributes.

The CNN model trained on the restaurant-level data achieves the lowest accuracy when compared to all other models. Depending on the attribute, the model can have more than 170 million trainable parameters, making it the largest model so far and requiring hours to train per epoch. However, the CNN trained on 100,000 samples of review-level data achieved the highest training accuracy (>97%) but performed the lowest on testing accuracy (~50%), a classic example of overfitting when the sampled reviews are not representative of all reviews. The results are not displayed in the table. When incorporating more than 100,000 samples, the CNN became too long to train in the allowed timeframe.

Something to keep in mind is that restaurant attributes are specified by restaurant owners, independent of review text written by customers. Some error in our model accuracy could be due to inconsistencies between restaurant owners' labels and what customers experience. Similarly, there could be a large variance in sentiment or overall experience in reviews for the same restaurant. Some reviews might have more positive sentiment towards *RestaurantsGoodForGroups* or *GoodForKids* while others may disagree.

In addition to accuracy, we proceed to analyze precision and recall metrics for our models. Precision measures how precise or accurate our model is; in other words, out of all of the attributes which we predicted to be true, how many are actually true. High precision corresponds to a low false positive rate, which is when we predict a restaurant to contain an attribute when the restaurant actually does not have an attribute. For example, the model predicts *GoodForKids* to be True, when in reality, *GoodForKids* is False for a restaurant. Recall measures out of the restaurants we classify as having an attribute, how many actually contain that attribute. High recall corresponds to a low false negative rate, which is when we predict a restaurant to not contain an attribute when the restaurant actually contains it. For example, the model predicts *GoodForKids* to be False, when in reality, *GoodForKids* is True for a restaurant.

5. Conclusion

We were inspired to help people find where they would like to eat, so our goal is to predict attributes based on review text. Overall, the Logistic Regression model using TF-IDF vectorized reviews performed best. That being said, model performance depends highly on the target attribute. Attribute prediction based on reviews is difficult due to high variation between reviews and finding relevant reviews. Future work could be used to generate attributes for sparsely labeled restaurants based off of review text. We can use clustering and semantic relations to recommend Yelp restaurants of a certain ambience. Ultimately, we seek to improve the Yelp customer experience and make the decision-making process of finding an appropriate restaurant more efficient, accessible, and enjoyable.

Next steps would include implementing and optimizing complex models such as BERT and transformer-based models and CNN to learn contextualized embeddings of reviews to predict ambience of a restaurant. We hope to fine-tune our models and apply it to make predictions on unlabeled restaurants. Other areas of future research extend to semi-supervised learning and auto labeling in the training process. In our case study we analyze a selected subset of binary attributes, but we hope to extend this to all restaurant attributes. Furthermore, we can create clusters of attributes and define standardized ambiences which our model can predict. A current constraint is to the long nature of our concatenated review texts per restaurant, so we hope to apply BERT to long-texts. "BERT is incapable of processing long texts due to its quadratically increasing memory and time consumption. The most natural ways to address this problem, such as slicing the text by a sliding window or simplifying transformers, suffer from insufficient long-range attentions or need customized CUDA kernels." [7]

References

[1] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.

[2] Fu, L., & Barrio, P. (2018). Distantly Supervised Attribute Detection from Reviews. NUT@EMNLP.

[3] Lin, Y., Shen, S., Liu, Z., Luan, H., & Sun, M. (2016). Neural Relation Extraction with Selective Attention over Instances. ACL.

[4] Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. COLING 2014.

[5]Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? CCL.

[6] Das, B., & Chakraborty, S. (2018). An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. ArXiv, abs/1806.06407.

[7] Ding, M., Zhou, C., Yang, H., & Tang, J. (2020). CogLTX: Applying BERT to Long Texts. NeurIPS.

[8] Shi, H., Ushio, T., Endo, M., Yamagami, K., & Horii, N. (2016). A multichannel convolutional neural network for cross-language dialog state tracking. 2016 IEEE Spoken Language Technology Workshop (SLT), 559-564.

[9] Brownlee, J. (2020, September 2). How to develop a multichannel CNN model for text classification. Machine Learning Mastery.

[10] Minaee, S., Cambria, E., & Gao, J. (2021). Deep Learning Based Text Classification: A Comprehensive Review.