Classifying Beer Styles Using an Interpretable NLP Model

Ian Naccarella
inaccare@stanford.edu

Wesley Olmsted
wolmsted@stanford.edu

GitHub: github.com/inaccare/Beer-Type-Prediction

1 Introduction

Mechanisms for understanding the factors behind the predictions made by machine learning models have become vital for the further advancement of such algorithms in real world applications. Currently, many such algorithms are opaque with limited ability to understand why they made a certain decision. This makes it difficult to trust them in high stakes situations, such as in medicine, as experts need to understand the reasoning behind a decision before moving forward with it. Furthermore, understanding the factors used by an algorithm in making a prediction could provide insight into the nature of the underlying data. Here we propose that interpreting a neural model used for classification can provide valuable insight into the intrinsic properties of the different categories.

We decided to test this thesis by applying the First Derivative Saliency method utilized by Li, Jiwei, et al in "Visualizing and understanding neural models in NLP" to the problem of beer type classification. Specifically, we used the RateBeer dataset provided on Piazza to learn how different types of beers tend to be described. For example, it is generally claimed that ales are more flavored and bitter than lagers. We wanted to see if this can be extracted from the set of reviews provided in RateBeer which we used for learning. In order to accomplish this, we first used the reviews provided to predict the type of beer being reviewed (i.e. Imperial Stout, Baltic Porter, IPA, etc). Then we attempted to determine which words in the review were most heavily weighted in making this prediction. Specifically, we built and investigated an interpretable neural model in order to understand which phrases or words help determine the style of beer. This then enabled us to determine which words are most significant in categorizing certain types of beer. In the future, we could also determine the impact that the various ratings have on determining the type of beer and thus will be able to determine which beers tend to be rated highly and lowly. Despite the somewhat limited practical applications available with this specific dataset, we believe that this work could easily be transferred to many other similar NLP problems. For example, wines tend to be defined by tasting notes which we could attempt to extract using this algorithm. Alternatively, restaurant reviews could be examined to determine how people view the various national chains.

2 Prior Literature

The literature on this topic details many different ways of building an interpretable neural model. While we eventually decided on using the aforementioned First Derivative Saliency approach, in this section we shall also detail a separate approach which we considered promising and which could potentially be implemented in future iterations, called a Generator Encoder model.

2.1 Generator Encoder

One of the most promising approaches for building an interpretable neural model, which we found during our literature review, came from the paper "Rationalizing neural predictions" by Lei, Tao, Barzilay, and Jaakkola [3]. In this paper, they investigate one type of interpretable neural model. Instead of a black box neural model which does not provide any rationale for the predictions it makes, this model provides justification. For example, in reviews, you can determine which parts of the reviews are responsible for the classification. The model is broken down into two separate but coexisting parts: the encoder and the generator. The encoder is responsible for taking in a sequence of words (called a rationale) and outputting a prediction. The generator is responsible for selecting text fragments from the review as candidate rationales for the encoder. The authors modified the loss function to penalize longer rationales so that the model learns the best, most concise rationales which can then be considered the words with the greatest impact on classification. Their model was able to achieve very high precision on different aspects of beer reviews and outperformed an SVM and a solid attention baseline.

2.2 First-Derivative Saliency

The approach we eventually decided on comes from the paper "Visualizing and understanding neural models in NLP" by Li, Jiwei, et al [5]. This paper outlines several strategies, borrowed from the world of computer vision, but focuses on First-Derivative Saliency. This strategy involves training a model utilizing word embeddings and then using the trained model to make a prediction while capturing the first derivative of this prediction, or score, with respect to each dimension of each embedding. This allowed the authors to provide saliency heatmaps (see Figure 1) for the various words in the review and thus determine which words impacted the classification the most. By doing this they learned that a Bi-Directional LSTM model is better than both a normal RNN and an LSTM with regards to filtering out irrelevant words, which influenced our decision to utilize a Bi-Directional LSTM.

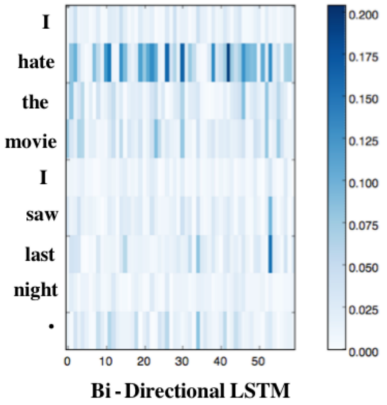

Figure 1: Saliency Heatmap from Li, Jiwei et al paper

3 Data

In this section, we will provide an overview of the dataset we used as well as the data processing needed prior to feeding it into our model.

3.1 Data Overview

Figure 2 provides an example of a review contained within the Ratebeer dataset. As you can see, there are 5 fields providing information on the beer and then 8 fields providing information on the various reviews of the beer with ratings from 1 to 5 for categories such as appearance, aroma, palate, and taste. For this project, we elected to only use the style field and the review field (marked by arrows in Figure 2) as we felt the review text would provide significantly more information than the ratings. However, future work could incorporate the various review fields to, for instance, determine whether Wheat Ales are deemed to have a better aroma than Doppelbocks.

3.2 Data Processing

Once we had examined our dataset, our next step was determining how best to process our data to optimize the performance of our classifier. Most papers in the literature

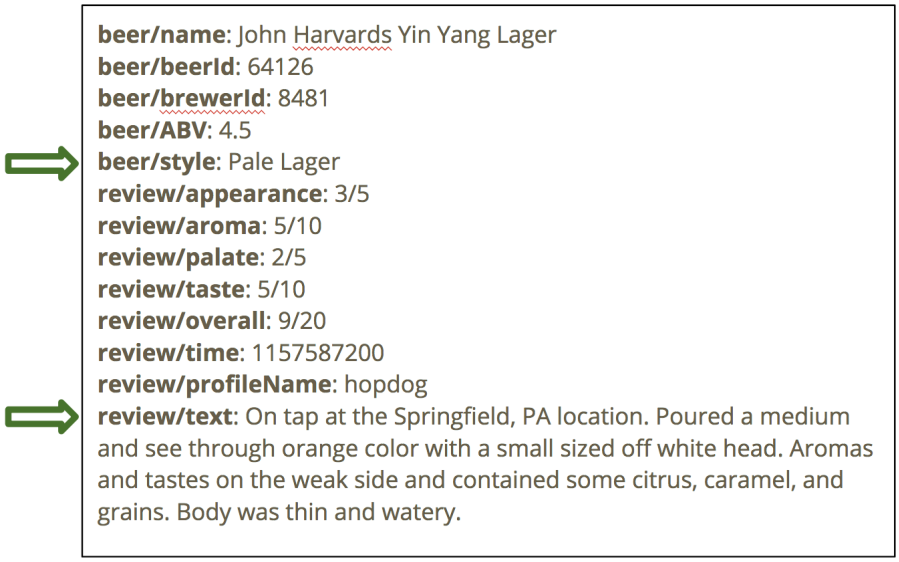

Figure 2: Example review from Ratebeer dataset

which work with the RateBeer dataset utilize 100-dimensional word embeddings so as to capture some sense to the meanings of the words in the reviews. Therefore, we utilized gensim's word2vec model to train word vectors on the reviews in our dataset, using a window size of 7, which we felt appropriate given that most of the reviews were 200-500 words long. This helped us gain insight into the context that the various words tend to be used in. This method was also preferable to utilizing pre-trained GloVe vectors because many of the reviews misspell words, which makes the context necessary and also means that the GloVe vectors would not have encompassed a significant portion of the vocabulary used in the reviews.

We also needed to determine where to cut off each review. The reviews varied greatly in length and we needed to pick a number of tokens that we would process in our RNN. If we picked a number that was too large, training would take very long, while if we picked a smaller number, we might miss out on key insights, especially since the end of a review tends to contain very important information summarizing the rest of the review. Based on Figure 3 (shown below), we

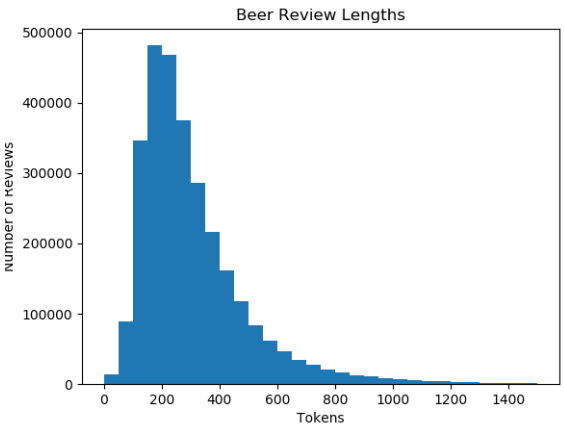

Figure 3: Histogram of Review Lengths

chose 400 as our max length. If the review was less than 400 tokens, we padded the rest of the sequence with a padding token, which was a randomly initialized vector.

3.3 Clustering

Prior to building our classification model, we also wanted to ensure that the reviews actually contained information pertaining to the type of beer being reviewed, thus ensuring a sound basis to the theory we were operating off of. To get this confirmation, we performed unsupervised learning to cluster

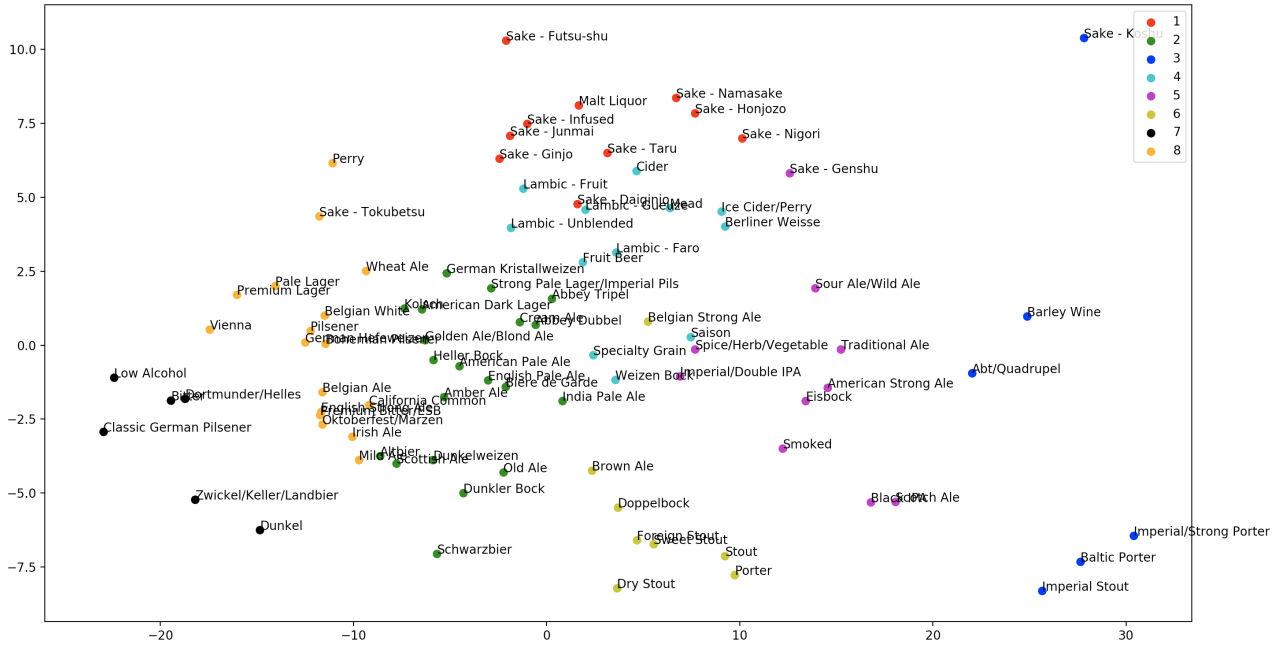

Figure 4: Clustered Beer Styles Using K-Means with pre-trained word embeddings

beers based on their type (i.e. Pale lager, Saison, Pilsener, etc.). First, for each review, we took the average of every word vector in the review as our basis for clustering. With these 100-dimensional vectors averaged over each review for each beer style, we formed 8 separate clusters using K-Means clustering. We then used PCA to reduce this down to 2 dimensions for visualization. The resulting clusters are shown in Figure 4. From the figure, we can see that purely based on the language of the reviews, certain beers are grouped together in ways that make intuitive sense. For example, sweet and fruity beers like mead, fruit beer, and cider are grouped together. Furthermore, lighter beers like pilsners and dortmunders are located in close proximity. Lastly, most of the stronger ales are grouped together as are many of the lighter ales and almost all of the sakes. This strongly indicates that the reviews do indeed provide information pertaining to the type of beer being reviewed and can form the basis for our model.

Given more time, we also could have taken counts of the descriptive words used in the reviews to see which words tend to be associated with certain beers. This could have acted as a type of baseline for our overall goal of determining which words are defining characteristics of certain types of beer.

4 Model

Prior to implementing neural interpretability, we first had to build a classification model to be interpreted. In this section we will outline our approach and the reasons behind the decisions we made.

4.1 Bi-Directional LSTM for Classification

Having confirmed that the beer reviews do contain information pertaining to the type of beer being reviewed, we proceeded to implementing a Bi-Directional LSTM model for classification. By examining the data, we determined that there are 89 different types of beer present in the dataset. We had initially anticipated performing classification based on the clusters determined by K-means, but our initial classification accuracies were high

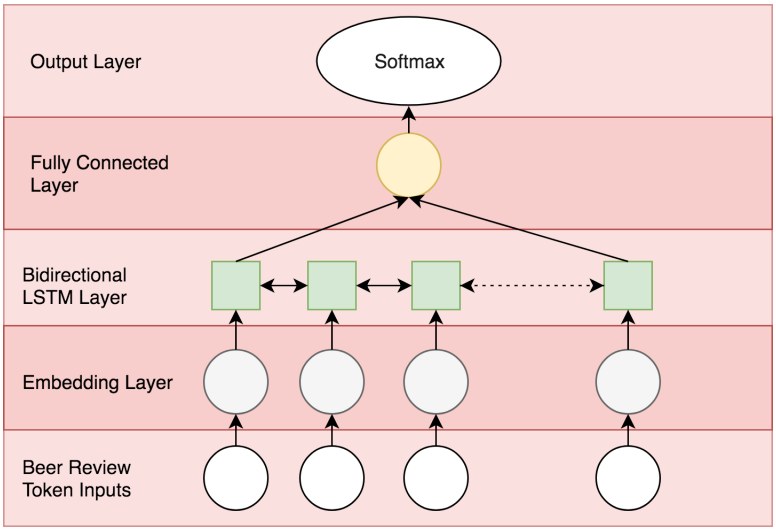

Figure 5: Outline of our Bi-Directional LSTM model

enough that we decided this was unnecessary. Thus, our model simply classifies the beer into one of these 89 different types.

The structure of our Bi-Directional LSTM is shown in Figure 5 above. The word vector representations of each word token in the review (mentioned in our Data Processing section) are fed into a Bi-Directional LSTM layer limited to 400 word embeddings. If the review contains fewer than 400 words then the remainder is padded with random vectors. The output of this LSTM layer is then fed through a single fully-connected layer and then into a softmax output function. The output is a softmax vector with length 89, corresponding to the 89 different beer types. The model then uses cross-entropy loss as its loss function.

4.2 Logistic Regression Baseline

We also built a baseline algorithm for classification using logistic regression as a benchmark to compare our Bi-directional LSTM to. For this model, similarly to the K-Means clustering, we averaged each word vector within each review to get its representation. We then performed logistic regression on that 100-dimensional vector in order to classify on 89 classes.

4.3 Classification Accuracy

As can be seen in Figure 6 below, our classification models were able to achieve fairly high accuracies considering it was a classification task with 89 different categories. Our best model achieved an accuracy of 44%, which is significantly higher than the 1% expected from random guessing. It is also significantly higher than the improved baseline achieved using logistic regression. We even attempted the classification problem ourselves using randomly selected reviews from the dev set and attempting to classify them based on our limited domain knowledge, achieving a 9% accuracy. It should also be noted that many of the reviews which our model misclassified were not very far off. For instance, our model would mistakenly classify a Stout as an Imperial Stout or mix up the various types of sake. Also, as can be seen from the low accuracy of the human classifiers, very rarely was the type of beer stated in the review.

Given these high accuracies, we felt confident that our model was extracting useful information from the underlying data and thus proceeded onto the next step of making it interpretable.

Model	Train Accuracy	Dev Accuracy
Baseline- Random Guessing	0.011	0.011
Baseline- Human Performance	-	0.091*
Baseline- Logistic Regression	0.064	0.064
LSTM	0.357	0.355
Bidirectional LSTM	0.442	0.440

* Performed on 33 random examples from the dev set

Figure 6: Classification accuracies of various models and baselines

5 Interpretability

5.1 First Derivative Saliency

As mentioned in the section detailing prior literature, First Derivative Saliency works by taking the gradient of the score function, which is used to predict the beer style, with respect to the word embeddings E. The equations we used are provided below. The absolute value of these gradients then tells you how much each word was weighted in making the final beer type prediction, which we are using as a proxy for determining which words the classifier deemed most important in classifying the review.

$$S_c(E) \approx w(E)^T E + b$$

$$w(E) = \nabla_E S_c$$

$$Sal(E) = ||w(E)||$$

In the Li, Jiwei, et al paper, the Saliency heatmaps are shown for each dimension of their 100-dimensional word embeddings. However, as seen in Figure 7, we decided to take the magnitude of this gradient and report that instead. This was decided upon because we saw little value in providing the derivative with respect to each dimension as the various dimensions in the word embeddings are not well understood.

6 Analysis

From the heatmaps provided in Figure 7, we can see that the classifier does indeed appear to be learning real word associations. For instance, the words “lemons” and “candy” were most heavily weighted in a review for Fruit beer while the words “piney”, “roasted”, and “chocolate” were deemed most important in a review for an Imperial Stout. We also performed independent verification online for a number of other types of beer and while this analysis is by necessity qualitative, our algorithm does appear to be emphasizing descriptive words commonly associated with the beer type being classified. Even more importantly, it appears to be ignoring descriptive words not commonly associated with the beer type. For instance, fizzy white heads are not generally associated with Belgian Strong Ales and our model puts very little weight on those words during classification.

Furthermore, the high classification accuracies achieved by our model give us additional confidence that it is truly learning word associations. In fact, we did not even have the time or computing resources needed to tune our hyperparameters, which we would expect to increase our accuracy even further. We also could have added more layers to our neural network which likely also would have

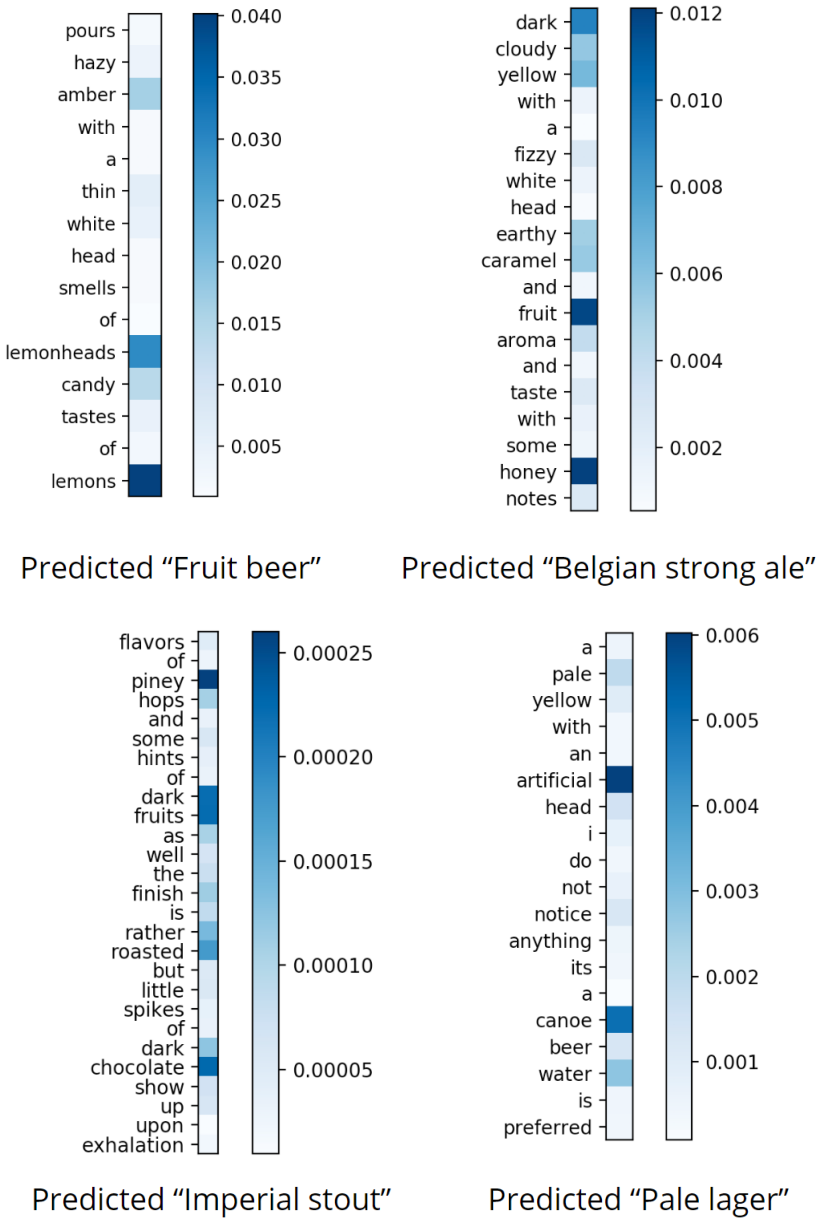

Figure 7: Saliency heatmaps for a various types of beer

increased our classification accuracy. Also, although we did not sanitize our dataset and ensure that there was an even distribution of the various beer types, by looking at the predictions made by our model we were able to determine that it does not appear to be favoring certain types of beer, thus artificially inflating its classification accuracy.

Another step would have been to perform classification based on clusters of beer type, such as by lumping together the different lagers, stouts, etc. However, this would have required significant research to determine how best to group the various beer types, though it most likely would have lead to very high classification accuracies. It also may have diluted the value of the saliency heatmaps for differentiating between the different beer types.

7 Conclusion

In conclusion, we believe that we have successfully proven our initial hypothesis that interpreting a neural model used for classification can provide valuable insight into the intrinsic properties of the different categories of classification. We were able to achieve accuracies above 40% when classifying reviews into 89 different styles of beer without even fully optimizing the model. We then saw through saliency heatmaps that our model did indeed appear to be learning real word associations and only utilizing the words which are most representative of the category.

Furthermore, while there may be limited applicability of our model when run on the Ratebeer dataset, we believe this could become a powerful tool for analysis on a variety of other datasets. For instance, if we were able to access a dataset containing reviews of various fast food restaurants, we could imagine running our model on those reviews to determine which words most closely describe the various restaurants. As an example, McDonalds might be ”cheap” and ”fried” while In-N-Out could be ”delicious” and ”quality” and Chipotle might be ”filling” and ”sick”. Thus, we can see that the power of this algorithm is limited only by the availability of data on which to run it, and that there will be innumerable applications for all sorts of interpretable neural models in the future.

8 References

1. McAuley, Julian, Jure Leskovec, and Dan Jurafsky. ”Learning attitudes and attributes from multi-aspect reviews.” Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, 2012.

2. McAuley, Julian John, and Jure Leskovec. ”From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews.” Proceedings of the 22nd international conference on World Wide Web. ACM, 2013.

3. Lei, Tao, Regina Barzilay, and Tommi Jaakkola. ”Rationalizing neural predictions.” arXiv preprint arXiv:1606.04155 (2016).

4. Li, Juncen, et al. ”Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer.” arXiv preprint arXiv:1804.06437 (2018).

5. Li, Jiwei, et al. ”Visualizing and understanding neural models in NLP.” arXiv preprint arXiv:1506.01066 (2015).

6. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. ”Why should i trust you?: Explaining the predictions of any classifier.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

7. Martins, Andre, and Ramon Astudillo. ”From softmax to sparsemax: A sparse model of attention and multi-label classification.” International Conference on Machine Learning. 2016.