Wine Recommendation System

Irene Ju
irenej98@berkeley.edu

Jeff Zheng
jzheng1@berkeley.edu

W266 Spring 2021, UC Berkeley School of Information

Abstract

In an effort to examine natural language processing approaches to classifying descriptive data, we present classic machine learning techniques (Logistic Regression) as well as deep learning algorithms (Neural Network and BERT) for identifying wine variety based on wine reviews. Using these various techniques, we compare various metrics for each model through a classification report. We find that our BERT model outperformed the two other models with a test accuracy of 78%. A further discussion into our error analysis reveals the top wines varieties that are misclassified and the potential sources of error.

1 Introduction

For beginners, wine can be very intimidating, especially since there are "dozens of different types of wine, each with their own ideal pairing foods" (Busch). As an initial consumer living in the era of the pandemic, going wine tasting is not an option, so many resort to online retailers. Having never tasted the wine before, they might make a purchase solely based on the popularity of the wine, rather than what suits their palate.

Our project strives to use descriptive words to help classify the variety of wine, so that it can help beginners choose the variety that best caters to their taste. There are four key wine descriptors: sweetness, acidity, tannin, and body, which will be the basis for the descriptions in our dataset. Sweetness deals with the amount of sugar content in the wine and can range from sweet, medium dry, off dry, and dry. Acidity is a mainly a descriptor for white wines, and can be said to be refreshing and crisp, or "sour" if overdone. Tannin refers to the natural substance found in grapes and can often be described as "bitter, causing a dry and puckery feeling in the mouth" (Busch). Lastly, is the body of the wine which is the weight and viscosity of the wine.

In order to address this problem, we implemented a deep learning approach to classification, first using a Neural Network, and then implementing a pre-trained BERT model with a linear classification layer. Neural networks are powerful tools for finding patterns in nonlinear data, which we found appropriate for our text classification text. Additionally, the technical innovation of Google's BERT leverages a bidirectional attention model to gain a better understanding of contextual language. We hope to leverage these models and assess their success in classifying descriptive text.

2 Background

Machine learning for classification continues to make great advancements in recent years, and Google's BERT, or Bidirectional Encoder Representations from Transformers, is one of the latest and most advanced developments for a variety of natural language processing tasks. As a highly evolved deep learning technique, we are interested in exploring how BERT compares to previous deep learning methods, such as neural networks. Deep Neural Networks have recently been at the forefront of machine learning due to their

ability to model complex relationships between nonlinear inputs and outputs, and have been very successful in a variety of classification tasks.

BERT is a new method of pre-training language representations that represents a milestone in natural language processing using neural networks (Google-Research). This innovative leap has produced state-of-the-art results for many NLP tasks, including text classification. BERT is a bidirectional model based on a transformer encoder architecture. This bidirectional attention-based model is pre-trained on two unsupervised tasks, Next Sentence Prediction and Masked Language Modeling. This allows us to take a pre-trained BERT model and retrain the BERT layer in order to fine-tune the model for downstream NLP tasks such as text classification.

Because of the contextual nature of descriptive text, we believe that BERT is a natural choice for our language classification problem. BERT's bidirectional attention algorithm will be able to help us extract high quality contextual language features from the wine reviews to more accurately classify the variety of wine being described.

3 Methods

We will focus on 3 models for our project. The first is a baseline multinomial logistic regression model using Bag-of-words (BOW), which creates vectors out of the descriptions in our data as features for our neural network (Real Python). The second is a Neural Network implementation of the BOW model. The third model is a BERT model using the HuggingFace PyTorch-Transformers model library.

3.1 Data

The dataset consists of 130,000 wine reviews scraped from the WineEnthusiast Magazine's website in November of 2017. It includes the wine name, variety, winery name and location, price, and review text and score. The dataset was retrieved from Kaggle (Zackthoutt).

3.2 Imbalanced Dataset

We observed that the collection of wine varieties were not evenly distributed. There were a total of 707 different wine varieties, however, we decided to focus on the top 10 most common varieties. Of the top 10 wine varieties, Pinot Noir was the most reviewed with 12,000 entries and Merlot had the least number of reviews with 3,100. We decided to down-sample the other nine classes to the same amount as Merlot for two reasons. The first was to reduce data imbalance between the different varieties of wine, so that each model received a similar amount of training data for each class. The second was to reduce the volume of data our models had to process, given our resource limitation in computing and processing power. This reduced our data size to around 24,000 entries total. (Appendix B & C)

3.3 Data Processing

Our data processing for logistic regression and neural networks was the same. We first pre-processed all the descriptions and used CountVectorizer provided by the scikit-learn library to create a vocabulary for the Bag-of-words models. The input data for our model was a feature vector containing information on the counts of words within the descriptions.

For the BERT model, we used the BertTokenizer from the pre-trained 'bert-base-uncased' to tokenize each of the descriptions. We added [CLS] indicating the first position and [SEP] which indicate the end of a sentence. Using the length of the longest description, we inputted [PAD] tokens to ensure uniform dimensionality of the descriptions.

For all three models, we used a 75-25 train-test split to evaluate the accuracy and see how well our model generalizes.

3.4 Model 1: Logistic Regression

For our baseline model, we decided to use multinomial logistic regression using the BOW feature vectors for classification of our wine varieties. This model will be used as a comparison with the more advanced models that involve deep learning, such as neural networks and BERT.

3.5 Model 2: Neural Network

Using keras from the tensorflow package, we built a basic neural network using the feature vectors from CountVectorizer as inputs. Each review was a 1x22339 input vector, for a Dense layer with 100 units using the ReLU activation function. This fed into our output layer, which used the softmax activation function for multi-class classification. The NN was trained for 150 epochs with batch size 32. (Appendix D)

3.6 Model 3: BERT

The BERT model, built using the HuggingFace transformer library, consists of an input layer for the token id vectors (1x194), that feeds into the BERT pre-trained base uncased model layer. We allowed the model to retrain the BERT layer. The classification layers include 0.5 dropout in each trainable layer, a Dense layer with 768 units using the tanh activation function, and a softmax output layer. This model was trained for 5 epochs with batch size 32. (Appendix E)

Model Details:

Both the neural network model and the BERT model used the Adam optimizer. Loss was calculated using cross-entropy loss.

4 Model Results and Analysis

4.1 Model 1: Logistic Regression

Figure 1 shows the confusion matrix produced from the baseline logistic regression model. The test accuracy for this model performed pretty well, reaching 75%. Merlot and Cabernet Sauvignon seem to be misclassified the most with each other as well as Sauvignon Blanc with Chardonnay.

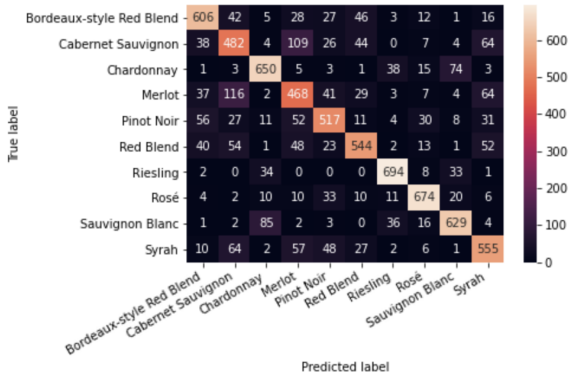


Figure 1: Logistic regression confusion matrix

4.2 Model 2: Neural Network

We ran 150 epochs for the neural network to see how the training loss and accuracy was after each iteration. From Figure 2, you can see that the training and test accuracies increase with each iteration. The test accuracy peaked at 77% with 125 epochs, then started to decrease, which means that

after 125 epoch, we might be overfitting our data. From Figure 3, the loss for both training and test continued to decrease with more epochs.

From the confusion matrix in Figure 4, the neural network seems to be having the same problem as the logistic regression in the misclassifications of Merlot and Cabernet Sauvignon as well as Sauvignon Blanc and Chardonnay.

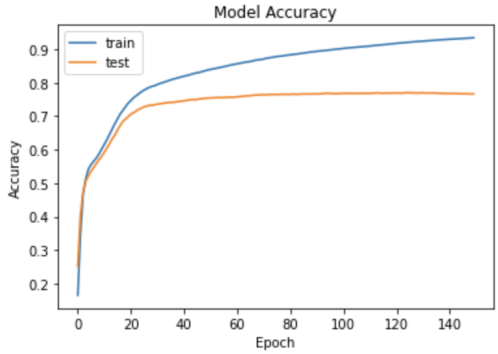


Figure 2: Neural network accuracy across epochs

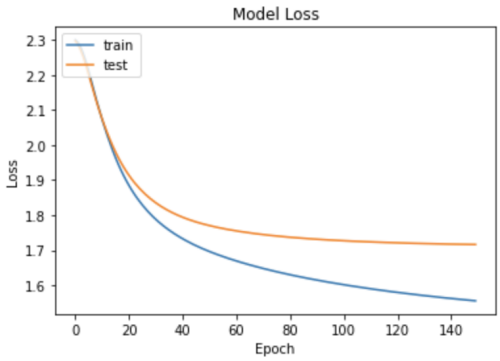


Figure 3: Neural network loss across epochs

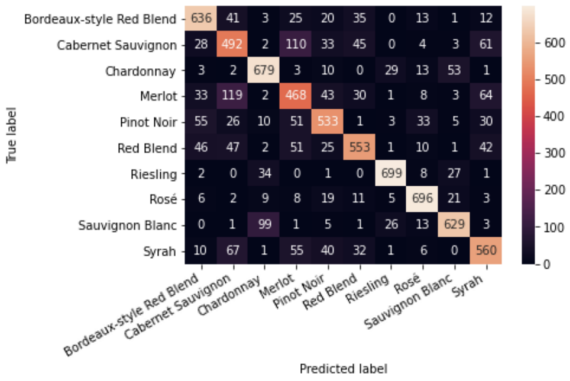


Figure 4: Neural network confusion matrix

4.3 Model 3: BERT

For our BERT model, we ran 5 epochs with each batch size of 32. Figure 5, you can see that the test starts off at a higher accuracy compared to the training. We believe our high dropout rate (0.5) may be hindering the initial training validation. The same reasoning applies for Figure 6 where the test loss starts at a significantly lower rate compared to the train.

The confusion matrix shown in Figure 7 contains the same problem as the two models above with the misclassification of the two red wines and two white wines. However, the correct classifications, shown on the diagonal, are higher than the other two models.

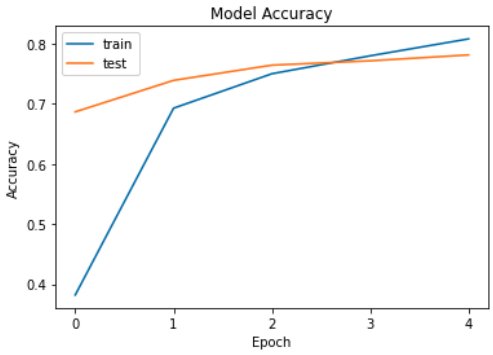


Figure 5: BERT accuracy across epochs

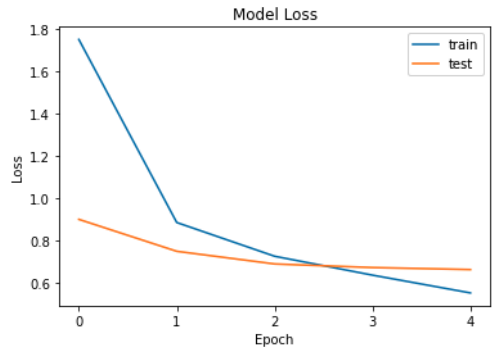


Figure 6: BERT accuracy across epochs

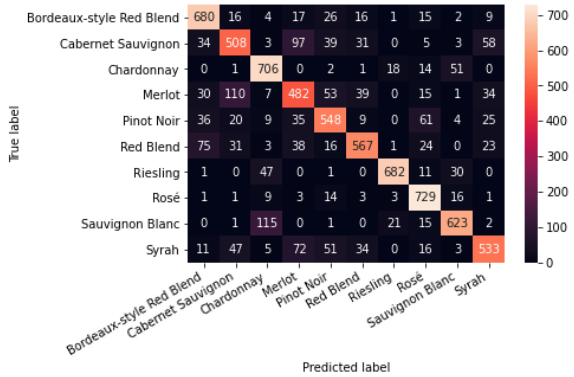


Figure 7: BERT confusion matrix

5 Discussion

5.1 Model Performance

Although we were limited by our computing resources and only ran the BERT model for 5 epochs, we see from our classification reports that the contextual language classification model outperforms both the neural network model and the logistic regression baseline model. Overall, the BERT model finished with an accuracy of 78% on the test set, compared to 77% by the neural network, and 75% by the logistic regression model. The BERT model also produces a better f1-score than the neural network for nearly all of the varieties. (Appendix F, G, & H)

5.2 Error Analysis

From all three models shown above, one common problem between all three was the misclassification of Merlot and Cabernet Sauvignon as well as Sauvignon Blanc and Chardonnay.

To better assess the performance of the models, we investigated the potential sources of error. In Figure 8, we printed the top 10 most common descriptive words for each variety of wine. Sauvignon Blanc and Chardonnay are both white wines, and the common or similar words used between the two varieties are: crisp, ripe, lime, citrus, grapefruit, and lemon. Although the types of fruit flavors are different in the two wines, they are all considered to be in the same family of citrus fruits, which makes sense why Sauvignon Blanc and Chardonnay are commonly misclassified.

```
Sauvignon Blanc counts:
        green: 1483
        citrus: 1443
        sauvignon: 1314
        blanc: 1189
        crisp: 1129
        drink: 1046
        lime: 1034
        grapefruit: 987
        fresh: 930
        ripe: 898
Chardonnay counts:
        apple: 3926
        chardonnay: 3315
        oak: 2877
        ripe: 2811
        drink: 2459
        lemon: 2298
        pear: 2193
        crisp: 2112
        rich: 2110
        vanilla: 1992
Merlot counts:
        merlot: 1287
        red: 772
        plum: 688
        oak: 646
        soft: 542
        spice: 493
        berry: 484
        ripe: 480
        dry: 478
        notes: 464
Cabernet Sauvignon counts:
        cabernet: 3170
        oak: 2624
        blackberry: 2430
        cassis: 1710
        drink: 1585
        chocolate: 1564
        ripe: 1526
        red: 1480
        plum: 1467
        currant: 1454
```

Figure 8: Top 10 counts of descriptive words

The red wines, Merlot and Cabernet Sauvignon, are the most misclassified out of all of the 10 varieties. Their common words include: red, plum, oak, and ripe. Furthermore, we sought out the descriptions for misclassifications of Cabernet Sauvignon for Merlot and noticed that often, descriptions of Cabernet Sauvignon

reference Merlot. One example of the description is shown below:

> "A healthy 20% of Merlot provides additional softness in this food-friendly, well-integrated wine, bright in crisp cranberry and mellowed oak and tannin. Well-made at a fair price, it's well worth the seek."

We see the names of wine varieties appear in the description of other wines, as a reference or comparison, which is very often misinterpreted. This wine is a Cabernet Sauvignon, and the descriptive words used that are found in Figure 8 are: merlot, soft, and oak. Although there are some mentions of the types of fruit used in the text, it isn't shocking that this description would be classified as Merlot, especially when the wine variety appears in the text.

Let's take a look at another example:

> "Aromas of tire rubber, plum and spice are followed sweet currant and cranberry flavors. The tannins bring some grit."

Descriptive words found in the text are: plum, spice, and currant. In reality, this description could serve as a description of either Merlot or Cabernet Sauvignon, but in this case, the true label was Cabernet Sauvignon, but got misclassified as Merlot. We believe that Merlot and Cabernet Sauvignon are often misclassified by our models because of how difficult it is for wine reviews to distinguish between these two varieties. Reviewers often have a difficult time tasting and describing a difference between them, which provides poor training data for our models (Admin).

6 Conclusion

Through our research, we have applied classic machine learning techniques as well as prominently favored deep learning techniques in order to classify descriptive-text. By comparing model performance through metrics such as accuracy, f1-scores, and confusion matrices, we were able to conclude that our bidirectional attention model performed better compared to a basic neural network and a logistic regression model. This task continues to support the growing field of natural language processing and the state-of-the-art techniques being developed to allow for complex feature extraction and modeling. We also conducted error analyses of our models in order to delineate potential aspects to continue to improve these techniques. In the future, we hope to include additional deep learning techniques, such as recurrent neural networks, in order to better assess which methods best apply to the classification of descriptive text.

7 Resources

7.1 Citation for Articles

Admin. “Admin.” Keenan Winery, August 24, 2017. https://www.keenanwinery.com/cabernet-sauvignon-versus-merlot/#:~:text=At%20first%20glance%2C%20many%20people,between%20Cabernet%20Sauvignon%20and%20Merlot.&text=Merlot%20tends%20to%20have%20a,up%20a%20slightly%20fruitier%20flavor.

Busch, Jack, Jack Busch, Jack Busch Jack Busch lives in the Pittsburgh area where he writes and edits for fun and money., Gordon Brown, Mike Henson, Ali Zagat, Stillman Brown, et al. “Wine for Beginners: An Easy Explanation of Different Wine Types.” Primer, January 28, 2020. https://www.primermagazine.com/2019/learn/different-wine-types.

Google-Research. “Google-Research/Bert.” GitHub. Accessed April 11, 2021. https://github.com/google-research/bert.

Real Python. “Practical Text Classification With Python and Keras.” Real Python. Real Python, March 19, 2021. https://realpython.com/python-keras-text-classification/.

Zackthoutt. “Wine Reviews.” Kaggle, November 27, 2017. https://www.kaggle.com/zynicide/wine-reviews.

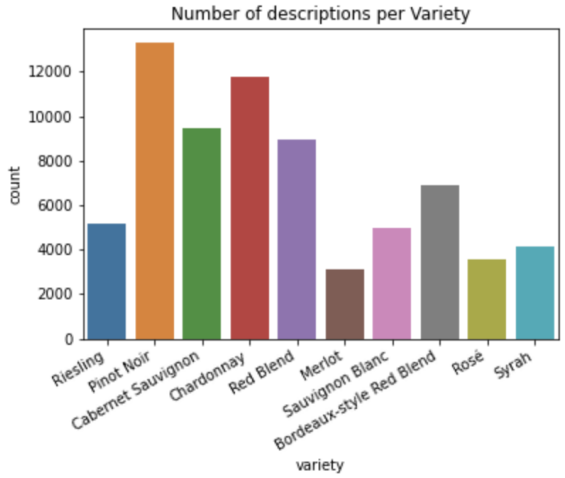
7.2 Resources for Code

Kshitijmohan. “Wine Recommendation System Based on BERT.” Kaggle. Kaggle, July 26, 2020. https://www.kaggle.com/kshitijmohan/wine-recommendation-system-based-on-bert.

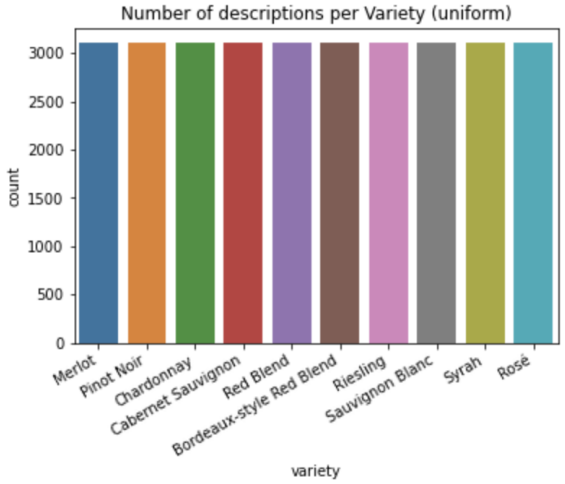
datasci-w266. “Datasci-w266/2021-Spring-Main.” GitHub, March 6, 2021. https://github.com/datasci-w266/2021-spring-main/blob/master/materials/Bert/BERT_T5_NER_2_3_030521.ipynb.

8 Appendix

A: Word cloud for descriptive words



B: Imbalanced varieties



C: Balanced varieties

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               2234000
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1010
=================================================================
Total params: 2,235,010
Trainable params: 2,235,010
Non-trainable params: 0
_________________________________________________________________
```

D: Neural network summary report

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_ids (InputLayer)       [(None, 194)]             0
_________________________________________________________________
tf_bert_model (TFBertModel)  TFBaseModelOutputWithPool 109482240
_________________________________________________________________
lambda (Lambda)              (None, 768)               0
_________________________________________________________________
dropout_37 (Dropout)         (None, 768)               0
_________________________________________________________________
dense (Dense)                (None, 768)               590592
_________________________________________________________________
dropout_38 (Dropout)         (None, 768)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                7690
=================================================================
Total params: 110,080,522
Trainable params: 110,080,522
Non-trainable params: 0
_________________________________________________________________
```

E: BERT summary report

	precision	recall	f1-score	support
Bordeaux-style Red Blend	0.76	0.77	0.77	786
Cabernet Sauvignon	0.61	0.62	0.61	778
Chardonnay	0.81	0.82	0.81	793
Merlot	0.60	0.61	0.60	771
Pinot Noir	0.72	0.69	0.70	747
Red Blend	0.76	0.70	0.73	778
Riesling	0.88	0.90	0.89	772
Rosé	0.86	0.86	0.86	780
Sauvignon Blanc	0.81	0.81	0.81	778
Syrah	0.70	0.72	0.71	772
accuracy			0.75	7755
macro avg	0.75	0.75	0.75	7755
weighted avg	0.75	0.75	0.75	7755

F: Logistic regression classification report

	precision	recall	f1-score	support
Bordeaux-style Red Blend	0.78	0.81	0.79	786
Cabernet Sauvignon	0.62	0.63	0.62	778
Chardonnay	0.81	0.86	0.83	793
Merlot	0.61	0.61	0.61	771
Pinot Noir	0.73	0.71	0.72	747
Red Blend	0.78	0.71	0.74	778
Riesling	0.91	0.91	0.91	772
Rosé	0.87	0.89	0.88	780
Sauvignon Blanc	0.85	0.81	0.83	778
Syrah	0.72	0.73	0.72	772
accuracy			0.77	7755
macro avg	0.77	0.77	0.77	7755
weighted avg	0.77	0.77	0.77	7755

G: Neural Network Classification Report

	precision	recall	f1-score	support
Bordeaux-style Red Blend	0.78	0.87	0.82	786
Cabernet Sauvignon	0.69	0.65	0.67	778
Chardonnay	0.78	0.89	0.83	793
Merlot	0.65	0.63	0.64	771
Pinot Noir	0.73	0.73	0.73	747
Red Blend	0.81	0.73	0.77	778
Riesling	0.94	0.88	0.91	772
Rosé	0.81	0.93	0.87	780
Sauvignon Blanc	0.85	0.80	0.82	778
Syrah	0.78	0.69	0.73	772
accuracy			0.78	7755
macro avg	0.78	0.78	0.78	7755
weighted avg	0.78	0.78	0.78	7755

H: BERT Classification Report