Deep Learning Application on Wine Rating Classification

Xiaoyun Hu Yang Jing

Abstract

With the development of various wording embedding techniques such as Word2Vec, GloVe, elmo and Bert lately, this paper looks at how to apply sentiment analysis to a specific field like wine industry that involves domain knowledge/terminology. We find fine-tuning Universal Sentence Encoder with additional 2 hidden layers boosts the network's learning about over 130k wine reviews and achieves 93.8% accuracy on wine rating classification.

1 Introduction

In this project, we will use wine critics' reviews to predict and classify wine rating. There are 4 ratings: acceptable, good, excellent, and superb. Description of wines can be very cryptic. They can have positive words such as "sweet" but labeled with a mediocre rating or they can have negative words such as "acidic" but labeled as high-end wine. We are interested to see how well an automated model can classify wine ratings by using word count based strategy vs using word embedding techniques. There doesn't appear to be studies done exactly in the wine rating context, but word count based strategies and word embedding techniques are widely applied in other fields, so we are able to leverage models from other studies and improve upon.

2 Background

Historically, sentiment analysis finds its use in consumer market for product reviews, marketing for knowing consumer attitudes/trends, social media for finding general opinion about recent hot topics in town and movie to find whether a recently release is a hit.

2.1 Challenges For Sentiment Analysis

Domain Dependency:
There are many words whose polarity changes from domain to domain. In the case of wine evaluation, critics follow the "Five S's of Wine Tasting" to fully appreciate each vintage. A wide range of vocabulary (refer to 3.1 EDA) is used to describe sight, swirl, snif, sip and spit.

Negation:
Handling negation is another challenging task. Negation can be expressed in subtle ways even without the explicit use of any negative words. Many of the wine critics express disappointment in this way.

2.2 Features For Sentiment Analysis

Lexical Based Strategy
refers to sentiment analysis that relies on lists of words and phrases with positive and negative connotations. For example, one method to quantify the semantic orientation of a sentence is as followed:

$$sentiment\ score\ =\ \frac{\#\ of\ Positive\ word - \#\ of\ Negative\ word}{\#\ of\ Positive\ word + \#\ of\ Negative\ word}$$

Many dictionaries of positive and negative opinion words were already developed. However in the case of wine analysis, this approach suffers from both of the two challenges discussed in section 2.1.

Count Based Strategy
Traditional (count-based) feature engineering strategies for textual data involve models belonging to a family of models popularly known as the Bag of Words model. This includes term frequencies, TF-IDF (term frequency-inverse document frequency), N-grams and so on. While they are effective methods for extracting features from text, due to the inherent nature of the model being just a bag of unstructured words, we lose additional information like the semantics, structure, sequence and context around nearby words in each text document.

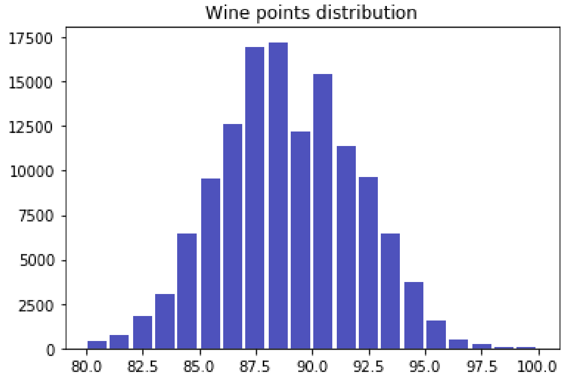
Word Embedding
refers to a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. They are pre-trained on a large volume of corpus through various tasks. Superior than the counting techniques, wording embedding preserves the semantic relationship among human language.

3 Methods

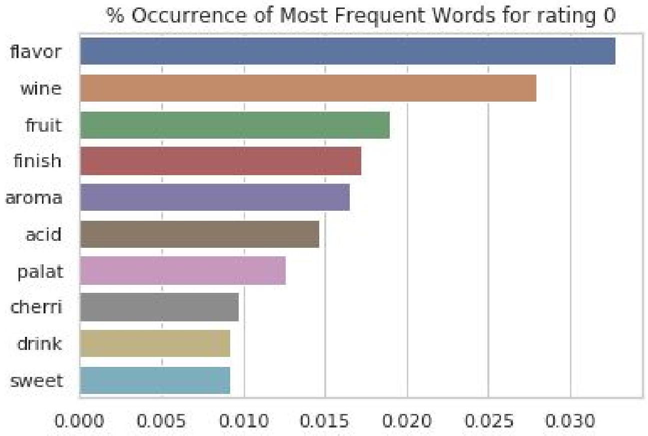
3.1 EDA

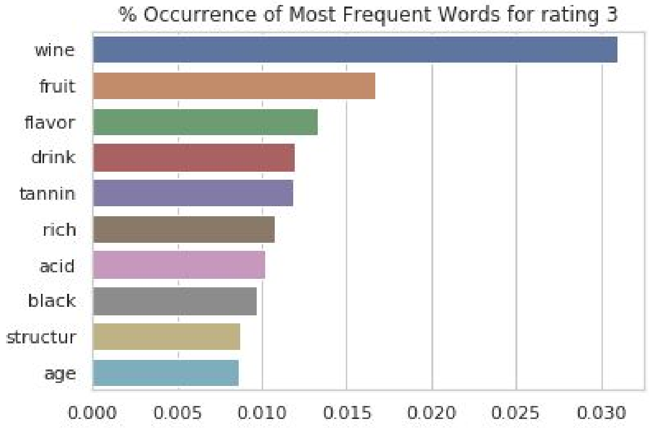
The dataset has over 130,000 data points. There are 13 columns but we will mainly focus on “description” and “points”. “Points” is our target variable. “description” and target variable both have complete entries without any missing values.

Points distribution, as expected, is bell-shaped with less counts on either end of the spectrum (very good or very bad wines). Most wines are in between 86 and 92 points.



We compared the top 10 most frequent words used to describe each class of wine. If you contrast rating 0 (bad wine) and rating 3 (good wine) below, “age”, “structure”, “black”, and “rich” only show up in high rating wines. Intuitively, good wines usually display qualities such as mature and rich in flavor, but interestingly, neutral words such as “structure” have a positive sentiment in wine description while “sweet” may not have the usual positive sentiment.





3.2 Pre-train Data Processing

WineEnthusiasts use a points scale ranging from 1 to 100 to rate their wines (1 being the worst, 100 being the best). Unfortunately, the website only posts positive reviews, which is why the range of the scores in the data set only range from 80 to 100. This paper converts a numeric measure to a categorical one following the below grouping. Data points in each class are not balanced. We will assign different class weights for training to address this issue.

Rating	Class	Review Example
80-85	Acceptable	A white this age should be fresh and crisp; this one is a bit dull and shows only faint bitter lemon flavors on the finish to go with tired apple aromas.
86-90	Good	Aromas of vanilla, char and toast lead to light creamy stone fruit and canned-corn flavors. It provides appeal but the oak seems overweighted.
91-95	Excellent	While it feels rich and round, this is also a wine with a crisp side to it. Acidity cuts through ripe pear, apricot and lychee flavors, giving a decidedly crisp balance. It's a wine that is already ready to drink.
96-100	Superb	The smoky, minerally aspects of the site show through even a thick veil of botrytis. This is intensely honeyed, filled with dried apricot and candied pineapple flavors, yet not without nuance. Incredibly sweet, but as balanced as a wine this unctuous can be, with a long, mouthwatering finish.

Furthermore, to clean reviews, we 1) removed common English words that don’t convey much information 2) converted all words to lowercase and their stem 3) cleaned accented character and special characters and digits.

3.3 TF-IDF Based Model (Baseline)

Github Model Notebook:
<https://github.com/tracyhxy33/w266_finalproject/blob/master/TFIDF%2BGloVe.ipynb>

Wine descriptions are fairly short, less than 100 words. The vocabulary of wine descriptions is also relatively small. TF-IDF is the simplest model we explored. It measures relevance instead of frequency. Each word is assigned a TF-IDF score. Train and test split is 50/50. We also used truncated SVD to reduce dimensionality. Vector matrix dimension went from (64985, 1448040) to (64985, 50). After inputs are vectorized, we explored Random Forest, Naive Bayes, and LinearSVC classify models to predict wine ratings. We don't expect TF-IDF based models to have high accuracy because we lose a lot of valuable information such as nearby words by using count based strategy.

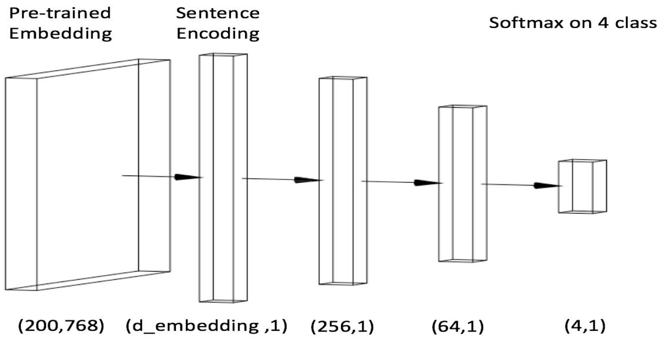
3.4 Word Embedding with Transfer Learning

Github Model Notebook:
<https://github.com/tracyhxy33/w266_finalproject/tree/master/wine_wordembedding_model>

The first word embedding technique we tried is using pre-trained GloVe with bidirectional LSTM. The network has one bidirectional LSTM layer with 32 units followed by a pooling, dense, dropout and another dense layer. Bidirectional means that the network will learn the text sequences in their original order as well as the reverse order in which the words appear. The classifier optimizes on accuracy. Accuracy does not differentiate between incorrect predictions, meaning that predicting a Class 0 as a Class 3 is no different from predicting a Class 0 as a Class 1. For future improvements, it would be worthwhile to modify the metric to differentiate "the level of incorrectness".

We then tried some of the new models out there that show tremendous results in a range of complex tasks such as inference or question and answering. They appear to have some level of language comprehension. Language models like NNLM (Neural Network Language Model), USE (Universal Sentence Encoder), BERT (Bidirectional Encoder Representations from Transformers) have shown that information learned from one dataset can be transferred to other datasets for specific tasks. A pretrained model takes input as a sentence and outputs vectors for each word/sentence. The vector it outputs is dependent on the context in which it occurs. This section applies the aforementioned models to wine sentiment classification and compare/contrast the effectiveness.

Model Architecture



This text classification model is built upon the sentence encoding layer provided by the 3 pretrained models. Two fully connected layers with size 256 and 64 followed by a softmax is added to predict wine review sentiment.

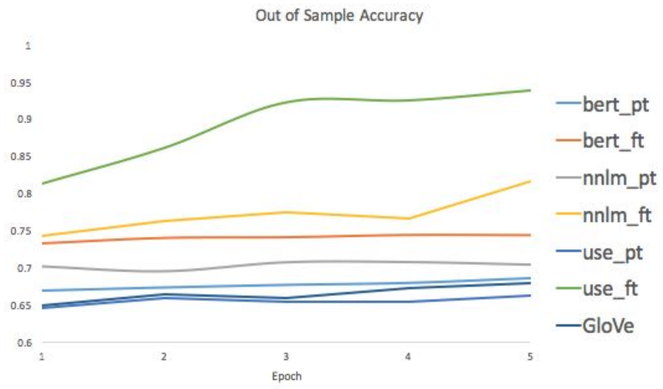
4 Results and Discussion

TF-IDF based models produced a reasonable baseline. Accuracy is the highest using LinearSVC, 64.1%. The train time for Random Forest and Naive Bayes is negligible, but the train time for SVM is significantly longer because of the curse of dimensionality. We downgraded to use LinearSVC instead.

TF-IDF Models	**Accuracy**	**f1-score**
Random Forest	61.4%	0.59
Naive Bayes	61.6%	0.57
LinearSVC	64.1%	0.57

Results using Word Embedding techniques:

WordEmbedding Model	**Finetune**	**Accuracy**
GloVe (100d) BiLSTM	FALSE	67.9%
NNLM (128d)	FALSE	70.4%
	TRUE	81.6%
USE (512d)	FALSE	66.2%
	TRUE	**93.8%**
BERT (678d)	FALSE	68.5%
	last 3-layer	74.3%



In all cases, finetuning improves the model accuracy. Pretrained models tend to stop learning after the first epoch. This is expected as finetuning enables the embedding layers to learn the domain knowledge for wine description. The best model for wine review turns out to be the fine-tuned version of universal sentence encoder as USE is trained on a number of tasks among which the main tasks is to identify the similarity between pairs of sentences. The authors note that the task is to identify "semantic textual similarity (STS) between sentence pairs scored by Pearson correlation with human judgments". This would help explain why the USE is better at identifying sentence similarity. While BERT is trained on two main tasks: masked language model where some words are hidden (15% of words are masked) and next sentence prediction where the model is trained to identify whether sentence B follows (is related to) sentence A. Neither of these tasks is specifically related to identifying whether a sentence is similar to another one.

Review Class	Acceptable	Good	Excellent	Superb
Acceptable	9971	953	8	0
Good	542	35074	1682	5
Excellent	3	739	15515	52
Superb	1	0	30	411

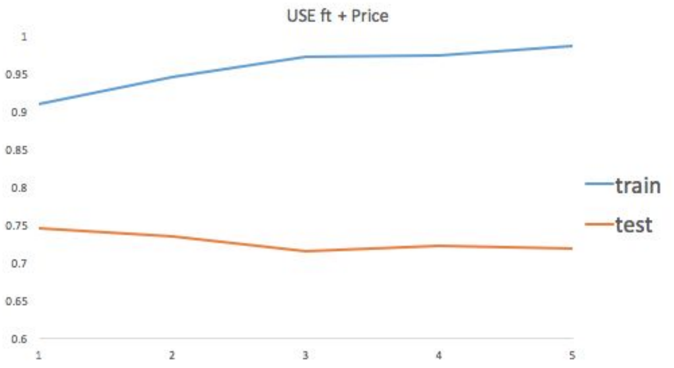
*Confusion matrix based on fine tuned USE

This is already a brilliant result given that neutral if not slightly negative words like "tannic" and "acidic" are used to describe high class wine. The relatively less accurate class is "Superb" where the model struggles between Excellent and Superb. A further investigation shows that review language becomes nearly indistinguishable within those two classes thus requiring an additional feature "wine price". As indicated in the following example, critics tend to give a higher rating to expensive wine though they express a comparable feeling in the review.

i) a $360 wine classified as Superb has the following review:
"Enormous tannins, dominant black fruit and a solid, dense structure. The wine, packed with dark fruits, dry tannins, very firm in character. With its huge tannins as well as fruit, this is a wine that really needs many years of aging."

ii) a $45 wine classified as Excellent has the following review:
"An enormously opulent wine, brimming with gobs of red cherries, currants, mocha, clove, cinnamon and pepper flavors. Impresses for the lushness and integrity of the structure. Brilliant wine, just gorgeous, a real crowd-pleaser. With its recent track record, Kynsi enters the front ranks of California Pinot Noir producers."

To test the conjecture that wine price provides additional information about rating, we augment sentence encoding with a numeric feature: wine price but keeping the rest of the network intact. However the result is not ideal.



The above training graph shows that with that additional price feature, the in sample fit surpass the best model (fine-tuned USE) reaching all the way to 98.7% while the out of sample accuracy lingers around 75%, which is a clear sign of overfitting.

5 Conclusion

We found that TF-IDF models are inferior than word embedding models without fine-tuning, and word embedding models without fine-tuning are inferior than word embedding with fine-tuning. TF-IDF models yield around 60% accuracy while word embedding models without fine-tuning yield around 70% accuracy. If we layer on fine-tuning, accuracy improves across all word embedding models. The reason is that words such as "sweet" and "acid" can have almost opposite sentiment in the context of wine verses in the context of movie reviews. By fine tuning the model, it was able to learn the context and improve accuracy significantly. USE with fine-tuning turns out to be the best model. It still confuses Excellent and Superb classes but including variable "price" in training only resulted overfitting.

References

Baseline Needs More Love: On Simple word-Embedding-Based Models and Associated Pooling Mechanisms: https://arxiv.org/pdf/1805.09843.pdf

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Lo¨ıc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In ACL. Association for Computational Linguistics.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. Context2vec: Learning generic context embedding with bidirectional LSTM.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP). https://nlp.stanford.edu/pubs/glove.pdf

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. Association for Computational Linguistics.

Hu M, Liu B. Mining opinion features in customer reviews. InAAAI, 2004; 4(4): 755-760.

Asghar MZ, Khan A, Ahmad S, Kundi FM. A Review of Feature Extraction in Sentiment Analysis. Journal of Basic and Applied Scientific Research. 2014: 4(3):181-186.