Predicting Country of Wine by Description Using Convolutional Neural Network

Yao Huang
Georgetown University
Washington D.C.
The United States
yh569@georgetown.edu

Abstract

This article discusses the project of a text classification task in Natural Language Processing. The project is about to predict the country the wine comes from with the description. The algorithm applied in this project is the convolutional neural network model with word embeddings on Tensorflow. In this project I will use the dataset containing over 130k records for wine reviews on Kaggle. The best model accuracy was 0.716, and when applying the model to the 100 testing descriptions, 76 of them were given the right predictions.

Keywords: text classification, convolutional neural network, reviews, tensorflow, word embeddings

1 Introduction

In natural language processing field, text classification has always been a fundamental but crucial task. It refers to the process of assigning tags or categories to text according to its content. In Natural Language Processing (NLP), it has many applications such as sentiment analysis, topic modeling, spam detection and so on. Usually, text classification uses supervised machine learning methods, as a labelled dataset contains text documents and their labels. This will be beneficial in training a classifier. Classifiers that have been implemented in solving text classification tasks include logistic regression, SVM, bagging model, boosting model, etc.

Different from traditional text classification tasks that are binary text classification such as email spam filtering (spam versus ham), multi-class classification usually includes more than 10 classes. Thus, training the classifier will have higher standards with the algorithm and the training datasets in order to reach acceptable accuracy. Neural network is an effective classifier. Earlier studies have implemented a series of experiments with convolutional neural networks (CNN) trained on top of pre-trained word vectors for sentence-level classification tasks (Kim Yoon, 2014). This study showed that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Since then, this has become a standard baseline for new text classification architectures.

In this specific project, I will provide our analysis and the text classification task based on a similar model of convolutional neural network (CNN). I will predict the country the wine is from by applying NLP techniques with the description of the wine. For example:

Description:
Elegance, complexity and structure come together in this drop-dead
gorgeous winethat ranks among Italy's greatest whites. It opens with
sublime yellow spring flower, aromatic herb and

orchard fruit scents.
The creamy, delicious palate seamlessly combines juicy white peach,
ripe pear and citrus flavors while white almond and savory mineral
notes grace the lingering finish.

Our ideal prediction given by the model should be: Italy.

This is a multi-class text classification task as in total, we have got 48 countries (labels) as the prediction results. In later discussion parts in this article, I will go into details about the methodology I applied, the datasets I used, and the results I got. The rest of the paper is organized as follows. Section 2 illustrates the algorithm/methodology, including the basic principles of convolutional neural network and word embeddings on Tensorflow. Section 3 explains the results, including the best model and the prediction results. Section 4 contains conclusions along with further discussion based on the whole project.

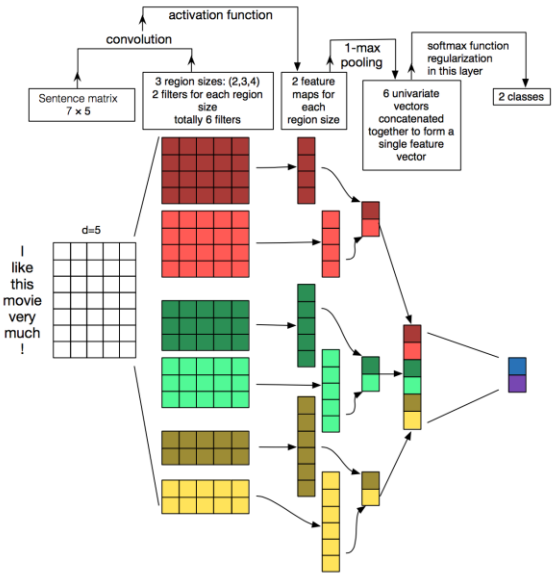
2 Methodology

2.1 Convolutional neural network

Convolutional neural network is now very broadly applied in the field of computer vision. CNNs were important for bringing major breakthroughs in image classification and are the core of most computer vision systems (Denny Britz, 2015).

What is convolutional neural network? CNNs are basically just several layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. Different from traditional feedforward neural network in which we connect each input neuron to each output neuron in the next layer, in CNNs we use convolutions over the input layer to compute the output. Each layer applies different filters and then combines their results. During the training phase, a CNN automatically learn the values of its filters based on the specific task.

In NLP tasks, the input are sentences or documents represented as matrix. In this specific project, our input is the description which includes several sentences. So how it refers to a matrix? Each row of the matrix corresponds to one token, typically a word. Each row is a vector representing a word. Both word embeddings such as word2vec and one-hot vectors can be applied to represent those vectors. The filters in a CNN can be used to slide over full rows of the matrix. A typical example for CNN on NLP tasks will be like the following:



2.2 Word embeddings on Tensorflow

In this project, I built the model with the word embeddings on Tensorflow.

To feed words into machine learning models, we need to convert the words into numeric vectors. A straight-forward way of doing this would be to use a one hot method. We convert the words into a sparse representation with only one element set to 1, the rest being 0. So there is two components to the Word2vec methodology:

1 word embedding: the mapping of a high dimensional one-hot style representation of words to a lower dimensional vector.

2 while we do this, we still maintain word context and meaning.

2.3 Datasets of wine reviews

This dataset I used in this project can be downloaded from Kaggle, credit to @zackthoutt. As described on the page, the dataset contains three files:

1. winemag-data-130k-v2.csv: contains 10 columns and 130k rows of wine reviews
2. winemag-data_first150k.csv: contains 10 columns and 150k rows of wine reviews
3. winemag-data-130k-v2.json: contains 6919 nodes of wine reviews

The data was scraped from WineEnthusiast in 2017. The dataset contains 14 columns (13 attributes). The descriptions of those attributes are listed below:

country	The country that the wine is from
description	A few sentences from a sommelier describing the wine's taste, smell, look, feel, etc.
designation	The vineyard within the winery where the grapes that made the wine are from
points	The number of points Wine Enthusiast rated the wine on a scale of 1-100 (though they say they only post reviews for wines that score >=80)
price	The cost for a bottle of the wine
province	The province or state that the wine is from
region_1	The wine growing area in a province or state (i.e. Napa)
region_2	Sometimes there are more specific regions specified within a wine growing area (i.e. Rutherford inside the Napa Valley), but this value can sometimes be blank
taster_name	Name of the person who tasted and reviewed the wine
taster_twitter_handle	Twitter handle for the person who tasted and reviewed the wine
title	The title of the wine review, which often contains the vintage if you're interested in extracting that feature
variety	The type of grapes used to make the wine (i.e. Pinot Noir)
winery	The winery that made the wine

In my project, I will only use columns ‘description’ and ‘country’. However, the work can be extended with other columns. I will talk about it in Section 4. To train the CNN model, I will use 100k random records from winemag-data_first150k.csv. Then I will test the prediction on randomly selected 100 descriptions from winemag-data-130k-v2.csv.

We will first do data processing with techniques such as lowercasing, lemmatizing as to get clean input. Then we implement the CNN model on the training dataset.

As we mentioned in earlier parts, in total, there are 48 countries considered as labels.

Albania	2
Argentina	4401
Australia	3355
Austria	2459
Bosnia and Herzegovina	4
Brazil	24
Bulgaria	59
Canada	166
Chile	4344
China	2
Croatia	84
Cyprus	22
Czech Republic	6
Egypt	3
England	9
France	17436
Georgia	39
Germany	2094
Greece	672
Hungary	169
India	8
Israel	553
Italy	19051
Japan	2
Lebanon	32
Lithuania	8
Luxembourg	6
Macedonia	11
Mexico	63
Moldova	49
Montenegro	2
Morocco	11
New Zealand	2170
Portugal	4463
Romania	105
Serbia	14
Slovakia	2
Slovenia	94
South Africa	1682
South Korea	4
Spain	6662
Switzerland	2
Turkey	52
US	49529
Ukraine	5
Uruguay	66

There are some observations from this country table of our training dataset. First, we can see that, though we have as many as 48 classes, a large proportion of these classes just contain a few training records. For example, Switzerland only got 2 records, while the US got 49529. What does it mean for classification? This will probably result in those labels with few records never being predicted since the model believes there is low possibility for a specific kind of wine to come from such countries.

3 Results

During the training phase, a directory was created, and the trained model was automatically saved to that directory.

The best model accuracy we got was 0.716. We applied this model to the testing data.

The input of the testing data was generated as json file including only ‘country’ and ‘description’ as for better implementation of fitting the model. There are many online convertors that can convert csv or excel files to json file. Below is one example of our input test data:

{

"country": "Portugal",

"description": "This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with juicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better from 2016."

}

We have in total 100 records for testing our best trained CNN model. The model gave a prediction represented as ‘new_prediction’. The output file was also a json file, including three elements:

'country': representing the right result
'description': representing the text input
'new_prediction': representing the predicted result given by the CNN model

The final results indicate we got an accuracy of 0.76 on the test set. In other words, among the 100 test data records, 76 of them were given the right prediction.

4 Discussion

This algorithm is not difficult to understand, however, there are many can be discovered from this result.

First, as we can conclude, 0.76 is of course not a perfect accuracy. Actually, according to the model accuracy of 0.716, if we extend the size of testing set, it is highly possible that the actual prediction accuracy will drop even lower. In text classification field, sometimes the accuracy can reach to 90% and above. However, considering the text documents we used in this project, one thing is, most of the descriptions do not contain much information related with the country the wine comes from, and this will bring trouble for the model to make predictions.

From the prediction results, we notice sometimes the description will mention the original country. This will be an indication, and the model could decide on whether this country mentioned is the true original country. This is one typical relation between the description itself with the label. However, there are counter examples. Let's look at the following description: "This shows a tart, green gooseberry flavor that is similar to New Zealand Sauvignon Blanc. Other notes include tropical fruit, orange and honey. Unoaked, with a splash of Muscat, this has commendable dryness and acidity."

Here the taster mentioned New Zealand, however, the actual original country of this wine is US. The model might not realize that. When it sees New Zealand, it will probably directly assign this wrong label.

So how can CNN model find the hidden relationship between the description and the label with 10k data records and try to give the best predictions? One probable solution is to try to connect some specific words to some specific labels. If a word is mentioned very often with some label, then this can be a good implication. In wine reviews, some words are mentioned often. For example, plum, berry, etc. It is possible that the country that has great quality products of these fruits is also the original country of this kind of wine made with these fruits. With more training data records proving this conclusion, the model can base the prediction on such findings.

To find connections between specific words and labels is a common way in text classification and proved to be very effective. In NLP tasks, for example, if "break" and "heart" appear together, then the sentiment will have a great possibility to be negative. The model can give this conclusion without doing more thorough detections on other words.

How can we improve the accuracy of the prediction results? One possible improvement is to combine information of description with other features, such as winery or variety. With information added, it would make more sense for the model to analyze and give predictions.

References

[1] G. Ou, Y. L. Murphey and L. A. Feldkamp, Multiclass Pattern Classification Using Neural Networks, in Proceeding of the 17th International Conference on Pattern Recognition (ICPR), 2004.

[2] P. Melville and R. J. Mooney, Constructing Diverse Classifier Ensembles using Artificial Training Examples, in Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), 2003.

[3] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for

modelling sentences. In Proceedings of ACL, Baltimore and USA, 2014.
[4] H. Wang, Z. Lu, H. Li, and E. Chen. A dataset for research on short-text conversations. In Proceedings of EMNLP, Seattle, Washington, USA, 2013.