Not All Categories are Created Equal

Ali Saffary, Anna Jeffries, McKenzie Shores

Abstract

We assess the performance of two neural networks that are trained using category-specific data versus less category-specific data. One neural network was a simple binary classification (positive or negative) whereas the other was for multi-class classification (positive, neutral, and negative). We anticipated gaining insights for modelling categorically-tagged textual data in an efficient manner. In conclusion, we postulate regarding the nature of what constitutes the 'best' category for training a neural network model.

Introduction

A question that arises when approaching product reviews generated from e-commerce is how best to model the data when that data spans a number of disparate categories, such as "Books" and "Furniture." Especially in light of the sheer *volume* of textual data produced from product reviews, evaluating the results in an efficient manner that still retains an acceptable level of accuracy is highly desirable.

In this project, we seek to utilize this principle in order to identify the "most general" category in the given data for training a model and evaluate *why* it is the "most general" by examining the vocabulary and accompanying weights (as opposed to unweighted words). We hypothesize that certain categories will contain reviews that are more descriptive and thus provide more meaningful data (after all, it is a general principle that the more context one has, the better one's understanding of language and its meaning).

We anticipate that the demographics of the reviewers will impact the extent and sentimental quality of their reviews.[1] For example, the category "Luxury Beauty" is most likely to contain reviews authored by women. Consequently,

[1] Due to the composition of our data, demographics can only be an oblique type of heuristic for us (or, at least, in the current stage we're in). The data contains the reviewer's screen name (e.g., "Mary S.," "Amazon Customer," "Trogdor") and it goes without saying that screen names are highly unreliable sources for determining demographics.

we would expect that "Luxury Beauty" will provide more sentiment[2] because women are more likely to use detailed language and emotive expressions. In contrast, in a category like "Automotive," we would expect most of the reviewers to be men who use more objective language that, comparatively, lacks in the amount of detail and personal sentiment.[3]

Ultimately, however, we remain interested in determining what makes a certain category the "most general." We also anticipate marked differences between a binary classifier and a multi-class classifier.

Pre-Processing

Our data set, Amazon Review Data, consists of 30 categories ranging from thousands to millions of reviews. For simplicity, we decided to limit the model to 10 categories. In order to select which 10 to include, we emphasized breadth over depth. It is more important to our hypothesis that our corpus covers a large range of content in order to train and expose our model to as many different types of reviews as possible. Additionally, we must consider the available computing power. While a large category may contain high quality reviews[4], it is worthless without the ability to process them.

The final 10 review categories are: Arts Crafts and Sewing, Automotive, Gift Cards, Luxury Beauty, Movies and TV, Office Products, Sports and Outdoors, Tools and Home Improvement, Toys and Games, and Video Games.

From each category we obtained 5,000 reviews that are split evenly between positive and negative reviews.[5] Ensuring an even split between positive and negative reviews is an essential step, as explained by Rashid and Huang. Subsequently, 20% of reviews were set aside as testing data from each category and 10% of the remaining data was set aside for development. All of the categories are combined for both the testing and development sets.

[2]By "more sentiment," we mean language that would lead to a more conclusive sentiment analysis, i.e. language that is easier for a model to learn and understand.

[3]We also acknowledge that it is possible that we discover the opposite to be true. Our presuppositions originate solely from the well-versed, pre-existing notion that women and men have distinctly different writing styles and the expression of personal sentiment differs therein. We also realize it could be easier to train a model favoring more objective reviews because the language construction will, generally speaking, be less complex and the vocabulary used less expansive.

[4]By "high quality" we mean reviews that contain the wealth of details and emotive expressions that would be most useful for training a model that relies at least in part on sentiment analysis.

[5]Assuming that Amazon reviews are skewed in the distribution of positive vs. negative reviews, our model would also be skewed without balancing. Because we have only conducted binary sentiment analysis so far, we selected only those reviews with one star (negative) or five stars (positive) initially. Two to four star reviews were also processed to be integrated into the model.

From the training data, we made two ordered dictionaries of all words are produced for the positive and negative sets. We then cut out everything but the top 100 positive or negative words (for the sake of simplification). During our initial project development we found that our review data created more features in our neural network than we anticipated. To combat this, we created a part-of-speech tagger so that we could eliminate non-essential words from our training data.

The tagger is created using a conditional random field model and was trained on the Penn Treebank data set. During development it achieved around 98% accuracy before applying it to our review training data. Because the reviews are not annotated, there is no concrete way to know how accurate the tagger is when applied to the Amazon Review Data, but from a manual inspection of the output, it seems to be fairly successful. After pre-processing, we have 36,000 reviews in the training set, 4,000 in the verification and development set, and 10,000 for testing.

Once the Amazon Review Data was processed, we also processed Google restaurant review data for out-of-sample testing. This data contained over 11 million reviews on a one-to-five star scale. We wanted to perform out-of-sample testing in order to experiment with the boundaries of the model's performance on data that was of a single broad category, uncurated by Amazon. The pre-processing for the Google reviews was fairly rudimentary; the data was not processed through the part-of-speech tagger nor was it split into any subsets. That said, all of the extraneous information[6] was removed, leaving only the text and corresponding rating for each review.

Model Creation

0.1 Binary Model

We wanted to create our initial model in a foreword-facing fashion (i.e., designed to make building upon it in the future a relatively simple task). With this in mind, we elected to pursue a multi-layer neural network using the Sigmoid activation function. We implemented four 1-dimensional layers, whose details are as follows:

Layer 1: 3,956 input nodes and 7,912 output nodes.

Layer 2: 7,912 input nodes and 3,956 output nodes.

Layer 3: 3,956 input nodes and 250 output nodes.

Layer 4: 250 input nodes and 1 output node.

[6]Unlike the Amazon data, the Google data set contained information like address, price, and pictures.

Activation: A Sigmoid activation function follows every layer.

Sigmoid is an ideal activation function because it does not lose information for the purposes of back-propagation. Additionally, as our target values are 0 and 1, they do not require any further indexing or pre-processing. We used binary cross-entropy to best fit the targets' binary format.

The most immediate complication that we encountered was the sheer quantity of features produced by CountVectorizer. From the 36,000 training samples, we obtained 39,566 features (i.e., unique words). As the number of features increases, the computations required of the neural network go up as well—exponentially, in fact, because each new node has to be connected to all nodes in the next layer. When the input layer grows too big, the following layers also have to grow alongside it to compensate for the extra load.

We have ignored features like numbers and stop-words (using regex and scikit-learn, respectively) as they don't convey any bias. Our current solution to this problem is to use scikit-learn's SelectPercentile which is a uni-variate feature selection method allowing the choice of the most important features. We chose just the top 10% of the 39,566 features, resulting in 3,956 final features. This not only allows each interaction of the training process to run faster, but it also allows the model to converge on its true accuracy using a lower number of epochs.

Once we addressed the features, we fed batches of the training data into our model. Each batch consisted of 100 samples with 10 epochs.

For each category, the model was trained and tested using two scenarios. In the first instance, 3,600 samples were used for training from the given category (e.g., 3,600 from "Beauty"). In the second instance, however, in addition to the 3,600 samples from the given category, 500 observations from each of the other categories were also added (e.g., 3,600 from "Beauty," 500 from "Automotive," 500 from "Arts & Crafts," etc.), making the total number 8,100.

Each of the two "versions" for each category were then tested on validation data, test data, and out-of-sample data (acquired from Google Maps, as previously mentioned). Our results are in the following section.

0.2 Multi-class Model

We also created a multi-class model which took the same Amazon reviews data as its input but also included the 3-star ratings as neutral reviews for classification. Another multi-layer neural network classifier was developed much like the previous one. The details of the model are described below.

Layer 1: 7,796 input nodes and 5,000 output nodes.

Layer 2: 5,000 input nodes and 2,000 output nodes.

Layer 3: 2,000 input nodes and 250 output nodes.

Layer 4: 250 input nodes and 3 output nodes.

Activation: Cross-entropy loss from PyTorch automatically applies Softmax activation for the final layer.

Since we are developing a multinomial model the best loss function to use is cross-entropy loss. This method is already provided by PyTorch. Said cross-entropy function also includes a built-in Softmax activation function. No other activation functions were chosen for this model as it stopped the model from learning for reasons beyond our understanding. This leaves us with no activation functions declared at the initialization of the model.

Like the Binary model, we used CountVectorizer to create our features for us as well as SlectPercentile to pick our top 20% most important features. This left us with 7,796 features to be fed into the model as input.

For each category, the model was trained and tested using two scenarios. In the first instance, 5,400 samples were used for training from the given category (e.g., 5,400 from "Beauty"). In the second instance, however, in addition to the 5,400 samples from the given category, 1000 observations from each of the other categories were also added (e.g., 5,400 from "Beauty," 1000 from "Automotive," 1000 from "Arts & Crafts," etc.), making the total number 14,400. Like above each "version" of the multinomial model was validated and tested against Amazon and Google review data.

Results and Analysis

0.3 Binary Classification

As demonstrated in the table below, the model results varied significantly between primary categories. The *minimum* validation accuracy for all models was 83% and the minimum testing accuracy was 76%.

In particular, the model trained exclusively on "Tools & Home Improvement" performed best for both training and validation accuracy, 99% and 97%, respectively. The *maximum* testing accuracy was 88% and was achieved by four of the more general models for the categories of "Beauty," "Movies & TV," "Office Products," and "Toys & Games." And, while ultimately secondary to our purposes herein, two models achieved an out-of-sample accuracy greater than 70% ("Arts & Crafts" and "Movies & TV").

As one may observe, the performance of the models aligns with the common sense notion that the more diverse the data is for training a model, the better results it will produce. In every single case, the more general models outperformed their more specific counterparts when it came to testing. However, it should be noted that there was no overlap between the best performers on the validation data and the best performers for the testing data. In theory, the validation and testing data would have similar distributions for the kind and

TABLE 1
Binary Classification Results

Category	Training	Validation	Testing	Out-of-sample
Beauty	0.97	0.86	0.84	0.65
	0.95	0.89	**0.88**	0.61
Automotive	0.98	0.83	0.81	0.55
	0.96	0.88	0.85	0.60
Arts & Crafts	0.96	0.91	0.82	**0.71**
	0.96	0.93	0.86	0.66
Gift Cards	0.96	0.87	0.83	0.63
	0.96	0.87	0.87	0.62
Movies & TV	0.96	0.89	0.82	**0.72**
	0.97	0.89	**0.88**	0.67
Office Products	0.97	0.89	0.84	0.61
	0.96	0.90	**0.88**	0.64
Sports & Outdoors	0.98	0.92	0.76	0.31
	0.97	0.95	0.87	0.60
Tools & Home Improvement	**0.99**	**0.97**	0.84	0.47
	0.97	0.95	0.87	0.58
Toys & Games	0.98	0.87	0.83	0.47
	0.97	0.89	**0.88**	0.43
Video Games	0.96	0.86	0.84	0.59
	0.96	0.87	0.85	0.67

Note: The initial rows filled with colour for each category represent the results for the model exclusively trained on the given category. The succeeding row represents the results of the model that was trained on other categories in addition to the primary category. **Bold** represents the results we have identified as significant for our analysis purposes.

type of reviews they contain. Perhaps this discrepancy is due to the fact that the size of the validation data is approximately half of that of the testing data (as previously mentioned, 10% of the data from each category was reserved for validation while 20% was set aside for testing).

Of what we will refer to as the "best" categories (e.g., the four that achieved the highest test accuracy), "Movies & TV" and "Beauty" are arguably two of the rational selections if one desires to train a more general model. They are both categories that lend themselves to highly descriptive and detailed language; as previously mentioned, women tend to write more than men and this certainly holds true in the "Beauty" category. And while the gender division for people who like to write movie and TV reviews is perhaps less clearly define, the overwhelming tendency is for lengthy descriptions and homilies regarding story plots and actors.

As for "Toys & Games" and "Office Products," there is no readily apparent explanation for their performance. Less specialised vocabulary is one potential

reason, but we don't believe it is the only one. "Toys & Games" can cover a large spectrum of items, from those bouncy toddler swing chair thingummies to Risk or Scrabble. It would've been helpful to have a clear idea of what products precisely do and don't fall under "Office Products." Does it include sundry electronic peripheral items (keyboards, mouse pads, etc.)? Printers and desks? Or is it more strict in the traditional idea of office products (e.g., paper, pens, staplers, etc.)?

Regarding the out-of-sample accuracies that exceeded our expectations, it is most curious to see that the best results came from specific category models as opposed to the more generalised ones. One may hypothesise that this is due to the broader scope of the categories in question, including their vocabularies.

0.4 Multi-class Classification

Not surprisingly, the multi-class model produced lesser results than the binary model. Introducing complexities to class labelling always make things more difficult, and we had a difficult time making a model that would achieve more than a measly 20% accuracy. So, while our results certainly seem poor on paper for this particular model, it is the best that we were able to produce.

Accuracies are highest in training, which is to be expected. However, once again, there are discrepancies between validation and testing accuracies. Unlike the binary model, though, there is an instance of overlap between the two ("Sports & Outdoors"). The precipitous drop between validation and testing accuracy for other categories is similar in degree to the drops we saw with the binary model.

Perhaps the most interesting observation regarding the multi-class model results is that the out-of-sample accuracies, while worse than in the case of the binary model, are not excessively worse than their binary counterparts. This suggests that, even though a model may have some success with out-of-sample testing, there is a ceiling, per se, to how well such a model can do without more intense development (which can lean dangerously towards the territory of trying to create a "universal" kind of classifier for all data that has review text and uses the 5-star system).

TABLE 2
Multi-class Classification Results

Category	Training	Validation	Testing	Out-of-sample
Beauty	0.92	0.61	0.58	0.57
	0.90	0.64	0.63	0.62
Automotive	0.93	0.66	0.59	0.62
	0.89	0.67	0.63	0.63
Arts & Crafts	0.88	0.70	0.56	0.52
	0.87	**0.74**	0.61	0.59
Gift Cards	0.86	0.59	0.56	0.54
	0.88	0.64	**0.64**	0.63
Movies & TV	0.95	0.71	0.59	0.59
	0.89	0.73	**0.64**	0.63
Office Products	**0.96**	0.68	0.57	0.60
	0.90	0.68	0.63	0.63
Sports & Outdoors	**0.97**	**0.74**	0.55	0.56
	0.89	**0.74**	**0.64**	0.64
Tools & Home Improvement	0.94	**0.74**	0.58	0.56
	0.90	**0.74**	0.62	0.60
Toys & Games	0.96	0.64	0.61	0.58
	0.90	0.64	**0.65**	0.64
Video Games	0.95	0.63	0.60	0.62
	0.87	0.64	0.63	**0.66**

Note: The initial rows filled with colour for each category represent the results for the model exclusively trained on the given category. The succeeding row represents the results of the model that was trained on other categories in addition to the primary category. **Bold** represents the results we have identified as significant for our analysis purposes.

Conclusion

Our study shows how important it is to understand your dataset prior to modelling and how you can simplify your inputs if you study it beforehand and understand which parts of your dataset are going to have the greatest impact on your results. When you have an understanding of the distributions of your dataset and their effects, then you can recognize and address any potential bias in your results.

We recognise that there are several shortcomings in our modelling. For one, we would have liked to expand our testing to other categories outside of the 10. And, perhaps more interestingly, we would have liked to develop a better multi-class model (which, we hypothesise, would have required larger training sets).