IMDB Sentiment Analysis

Sergio R. Robledo

Springboard

Author Note

You can contact me at sergiorobledo2014@gmail.com for further information.

Abstract

Using the IMDB dataset [1] which consists of 50,000 labeled movie reviews, we hope to accurately predict the sentiment of any given movie review. Initial data preprocessing will be done, in order to aid in the exploration and visualization of the dataset; data exploration will be followed up by inferential statistics to further confirm the validity of the observations made in the exploration stage of the dataset. Finally, the BERT [2] model will be trained on half of the labeled movie reviews to classify any given movie review as expressing either a positive or negative sentiment. The fine-tuned BERT model was able to achieve a top 10 state-of-the-art test accuracy of about 94.93%.

Keywords: NLP, BERT, Binary Classification, Inferential Statistics, Data Wrangling, Exploration, Visualization, Machine Learning, Sentiment Analysis, IMDB, Python, Deep Learning, Neural Networks.

IMDB Sentiment Analysis

Sentiment analysis is one of the main areas studied by the Natural Language Processing (NLP) field, and the accuracy metric achieved is constantly being pushed to new limits as technologies and algorithms are developed. In this project, we will be using one of the most recent breakthroughs in NLP to predict the sentiment of the IMDB movie reviews. However, before delving into the BERT algorithm, we have to understand the motivations driving the need for a classifier, as well as the various processes that go into understanding the data to be modeled.

1. Project Proposal

Given the high volume of movie reviews written every day, there is no way a team of people can be assigned to correctly classify the rating of each movie review; therefore, an automated machine learning model must be developed in order to provide an accurate representation of a movie's rating given only the text inputs for that movie's reviews. The more accurate the classifier, the more the rating website, in this case IMDB, will be visited, because users will learn to rely on the site's assigned rating to any given movie. The final rating will be derived as the proportion of positively classified movie reviews to the total number of reviews that movie received; for example, if 8/10 movie reviews are classified as positive, the movie will receive a rating of 0.8. Therefore, it is crucial that the learning algorithm achieves the highest accuracy possible.

1.1. Acquisition of data?

The IMDB dataset [1] consists of 25,000 training labeled movie reviews which will be used to train the learning model, and the remaining 25,000 testing labeled movie reviews will not be fed to the learning model until it has been fully trained and will be used to assess the performance of the model; all the movie reviews are labeled as expressing either positive or negative sentiment.

1.2. Solution Brainstorm & Deliverables

The initial proposal of the project included data wrangling and a final deep learning model able to correctly predict the sentiment of a given movie review more than 85% of the time. The culmination of the project will result with the complete model code and a PowerPoint slide deck discussing the results.

1.2.1. Model Code.

The model code will include the data wrangling, exploration and visualization done on the IMDB dataset. It will also include all the code used to train the BERT model, as well as the results obtained after training the model. If no further training is desired, the model checkpoint at which the model was able to achieve the final accuracy will be provided, so future movie review classifications can be done without any additional training of the BERT model.

1.2.2. PowerPoint Slide Deck.

The PowerPoint slide deck discussing results will discuss any additional requirements needed if future training was desired; for example, the need of Google's cloud platform TPUs. The most critical steps taken from acquisition of data to the training of the model will be summarized and explained. Finally, the model's performance, i.e. accuracy, will be included in the presentation along with possible future steps that could improve the model's performance.

2. Data Wrangling

Data wrangling is essential in any data science and machine learning project, since it is oftentimes the case that the data acquired is not ready to be analyzed or train a model. Two stages of data wrangling was done on the IMDB dataset, the first data wrangling stage was done for the sake of exploration and visualization of the dataset, and the second data wrangling stage was done to get the data into a format which could be used by the BERT model to train.

2.1. Data Wrangling: Exploration & Visualization

In order to get useful observations and conduct inferential statistic tests on the IMDB dataset, we have to convert movie reviews into a similar type of format. The preprocessing steps needed to prepare the dataset for data exploration and visualization includes the lowercasing of words; removal of regular expressions, stop words and uncommon words; and tokenization of reviews.

2.1.1. Lowercase.

In order to reduce the variation in between movie reviews due to capitalization, we will first lowercase all words in the movie reviews, so words that would otherwise be counted as two different words due to capitalization, will now be counted as the same word. For example, treating "The" and "the" the same will aid in the removal of common words that do not add sentiment by making their frequency level appear higher than other words, and in turn help us find the words that are used very often but provide no sentimental value to the review.

2.1.2. Regular Expression Removal.

Another technique that will help reduce the variation of words used within reviews is the removal of regular expressions; regular expressions include all forms of punctuation as well as other values found within reviews that are not established under the English language. This step further aids the data exploration process by treating similar words as one word; for example, if the word "ugliest" is used to end the sentence, it will contain a punctuation symbol attached to the word and will be treated as if it's completely unrelated to the actual word, "ugliest", so by eliminating all punctuations before the tokenization of the words, we ensure that words don't get incorrectly classified as different words due to slight variations between reviews.

2.1.3. Word Removals.

When comparing the differences between positive and negative reviews and what words are most often found in each one, we want to make observations that actually add to our understanding of the reviews; therefore, we want to remove any words that do not provide sentimental value to the movie reviews and will only make the exploration of words that actually contain sentimental value harder to do. In order to keep only the most relevant information within reviews, we will remove all English stop words and other uncommon words.

Stop Words.

English stop words include words such as "the", "him", "her", "or" and "they", among others. These words are oftentimes used very frequently due to the nature of the English language but do not add any sentimental value to a movie review; therefore, when making observations of what words are most frequently used in reviews, stop words will often appear at the top of the list and block other words that do add important sentimental value to reviews but do not appear as often. In order to prevent this, English stop words will be removed from the movie reviews.

Uncommon and Common Words.

To make our observations more statistically significant, we will remove uncommon words that might lead to observations whom were likely the result of chance; we will classify uncommon words as words that appeared in less than 100 of the training reviews. Further, we will remove any common words that were not pruned out by the English stop words filter and do not add sentimental value to the review; for example, the word "movie" was not removed in the English stop word removal process, and it is used very frequently within reviews.

2.1.4. Tokenization.

Finally, once all the previous steps have been applied to the dataset, we can now begin our exploration of the dataset. In order to make the exploration much more streamlined, we have to a tokenize the movie reviews; tokenization entails the mapping of words to an assigned index. Tokenization allows for word counts and document counts of all the words in the reviews to be retrieved and analyzed with ease.

2.2. Data Wrangling: BERT Model

Data exploration is only one critical part of the data science process; training the machine learning model is another crucial part in developing a successful machine learning model, and oftentimes the data acquired is not compatible with the training process of the model. The BERT model cannot be trained on raw text data and many of the preprocessing steps taken to prepare the data for exploration will need to be done to convert the raw text data into something the BERT model can train on; however, there are slight variations between the preprocessing steps of data exploration and model training. Preprocessing the data for the BERT model does not need the removal of regular expressions, uncommon words, or other non-sentimental words; however, it will need the lowercasing, splitting of regular expressions, and tokenization of all movie reviews.

2.2.1. Lowercase.

The lowercasing applied to get the data ready for model training is the same process as the lowercasing applied to the data when getting it ready for exploration.

2.2.2. Punctuation Splitting.

Even though regular expressions, i.e. punctuation, do not need to be removed for the training of the BERT model, they do have to be separated from the rest of the word and treated as a word of its own; for example, the word “you’re” would be turned into “you ‘ re” after punctuation splitting is applied where the original word is now equivalent to three different words.

2.2.3. BERT Tokenization

Tokenization for BERT is also slightly different from the simple word count tokenization applied to the data during the exploration phase, since BERT tokenization uses WordPiece tokenization to represent a sequence of words; WordPiece tokenization splits words according to

their root word. For example, the word “homeless” would be converted to “home ##less” where the original word is now two separate words, i.e. tokens.

3. Data Exploration, Visualizations & Inferential Statistics

Once the data wrangling preprocessing steps needed to perform data exploration are completed, the data is ready to be analyzed, explored and visualized. Since we only have text data, IMDB’s data exploration will be mainly focused on analyzing word counts and looking for what sorts of things differentiate positive and negative reviews.

3.1 General Observations

The first part of data exploration will be done on all the movie reviews as a whole, with both positive and negative reviews grouped together. We will look at the average review length and the most commonly used words within all reviews.

3.1.1 Review Length.

The average review after preprocessing is about 94 words in length; however, more than half of the reviews are 71 words or less in length. Whenever the median is significantly lower than the average, you can assume the distribution of review lengths is very positively skewed, since the outliers with longer review lengths pull the average value a lot higher than the median; this observation is further confirmed by the histogram plot of review lengths and can be seen below.

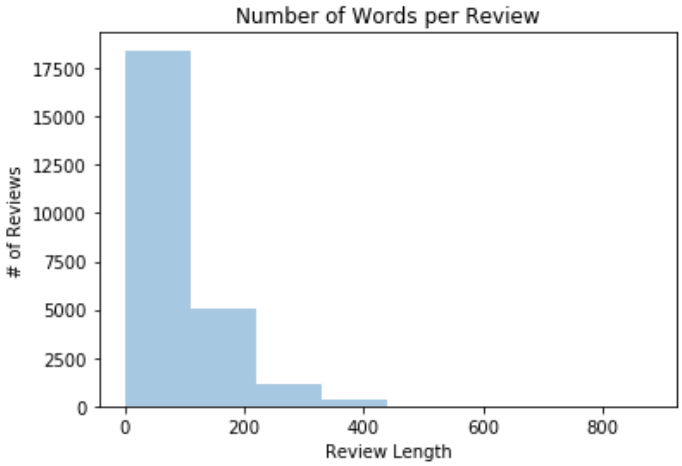

Figure 1: Review Length Distribution

3.1.2. Word Counts.

After tokenization, it is really easy to do word counts on a large dataset, and after applying preprocessing and tokenization to the IMDB dataset, we are able to see some of the

most frequently used words in all of the movie reviews, regardless of sentiment. Some of the top 10 most used words in the reviews include the words “one”, “like”, “good”, and “really”; by themselves, these words somewhat provide sentiment value. For example, the word “good” could be associated with a positive sentiment; however, given that these words appear very frequently in both the positive and negative reviews, one has to wonder how or why the “good” would be used in negative reviews. The word “good” could be used in a positive review to describe a movie being “really good” or in a negative review to describe the movie as “not good”. This leads us to believe that a stand-alone word cannot provide all the necessary information needed to classify a movie review as either positive or negative; context will be equally if not more important for the classification of movie reviews, and this conclusion will prove useful when deciding what the correct machine learning model is for the current dataset.

3.2. Positive & Negative Observations

Now that we have explored the IMDB dataset as a whole, we can move on to grouping positive and negative reviews with the purpose of making observations that will give us a clearer picture of what separates a positive and a negative review. We are going to first check if there’s any significant variation between both types of reviews with regard to the measurements done on the whole dataset, e.g. review length. Then we will select a few polarizing words such as “love” and “hate” and record the differences between the number of times those polarizing words appear in the positive and negative reviews; this will allow us to see whether individual words add any value towards the classification of a given review, since there should be a clear difference between the positive and negative reviews’ word counts of polarizing words.

3.2.1. General Observations Redone.

Now that we have separated positive and negative reviews into their own groups, it is worth exploring whether or not observations made earlier for the entire dataset still hold true for the grouped reviews.

Number of Reviews.

There are 12,500 positive and 12,500 negative reviews; a balanced dataset containing equal number of both types of reviews is very important when training a classifier, because an unbalanced dataset indirectly influences the weights of the trained classifier. For example, if there were 12,500 positive reviews and only 500 negative reviews, the classifier would be heavily in favor of classifying any random review as positive, since it got trained to incorrectly

believe that the probability of making a correct prediction increases if it predicts the random review to be positive.

Review Length.

The review length measurements for both groups is surprisingly similar to the review length measurement of the whole dataset; positive reviews had an average review length of 95 words with half of the reviews being 71 words or less, and negative reviews had an average length of 94 with half of the reviews containing 72 words or less. Therefore, it can be observed that review length will not help in determining whether a review is positive or negative.

3.2.2. Polarizing Word Counts.

In order to find out whether individual words provided assistance in differentiating positive and negative reviews, a few polarizing word counts were recorded for both datasets; polarizing words are words that contain a very high level of sentiment within them, and the polarizing words recorded are the words “love”, “awesome”, “hate”, “boring”, “exciting” and “horrible”. Below is a histogram and a table denoting the number of reviews each word appeared in.

Positive Reviews	
love	2825
awesome	304
hate	301
boring	300
exciting	279
horrible	152

Negative Reviews	
love	1614
boring	1192
horrible	842
hate	395
exciting	194
awesome	108

Figure 2: Polar Word Counts

As it can be seen above, there are some clear distinctions between positive and negative reviews with regard to the number of times polarizing words appear within their reviews; for example, the word “love” which clearly has a positive polarization appears in 1,211 more positive reviews than negative reviews. On the other hand, the word “boring”, which contains very negative sentiment for a movie, appears 892 more times in a negative review than in a positive review. Therefore, it can be concluded that there is indeed a difference between positive and negative reviews regarding the frequency of words. To confirm that these observations are not due to chance, we will run inference statistics on the some of the observed values.

Inferential Statistics.

Inferential statistics allow us to test observations and conclude with a certain level of confidence that what has been observed is statistically significant and not due to chance. Below we will apply the bootstrapping method to figure out whether the difference between positive and negative reviews regarding polarizing word counts is not due to chance; more specifically, we will test the observations made for “love” and “hate”.

“love”.

After simulating 100,000 circumstances where positive and negative reviews contain the word “love” equally among their reviews, there were 0 out of 100,000 situations where the difference between positive and negative reviews containing the word “love” reached the observed 1,211 review difference. Therefore, it can be stated with more than 99.99% confidence that there is indeed a difference between positive and negative reviews with regard to the number of times the word “love” can be observed within them; in fact, more than 99% of the 100,000 simulations observed a difference of 155 reviews or less for the word “love”.

“hate”.

After simulating 100,000 circumstances where positive and negative reviews hypothetically contain the word “hate”, there were 16 out of 100,000 simulations where the difference between positive and negative reviews containing the word “hate” reached the observed 94 review difference. Therefore, it can be stated with more than 99.9% confidence that there is indeed a difference between positive and negative reviews with regard to the frequency to which the word “hate” is used within them.

4. BERT Model

The final step in the data science process is the training and evaluation of the machine learning model. Given the insights gained in the data exploration aspect of the project, we can safely assume that while some highly polarized words do help differentiate between positive and negative reviews, one of the most important aspects the machine learning model will have to take into consideration is the context of a given review; therefore, when choosing the machine learning model we have to make sure that the model was optimized to understand the effect context has on the overall sentiment of the movie review. After trying out different model architectures and techniques for representing text sequences, I arrived at the BERT model. The BERT model makes use of the Transformer algorithm [3] to learn the context of a word within a sentence and overall relationship between two sentences; we will exploit BERT's ability to predict whether or not two sentences are correlated by replacing the model's original sentence correlation prediction with the sentiment classification of the labeled reviews. BERT is composed of two main training stages referred to as pre-training and fine-tuning; we will discuss each stage in the following sections. Further, BERT comes in two main forms and which one you choose depends on how much RAM you have access to; using one of Google Cloud Platform's TPUs which had access to 64 GB of RAM, I was able to use the bigger BERT model of the two known as BERT-Large.

4.1. BERT-Large Specifications

Further, BERT-Large has a "cased" and an "uncased" version of the model where the "cased" model does not require the text input to be lowercased and the "uncased" does. The main difference between the smaller BERT model, BERT-Base, and BERT-Large is the size of the model; BERT-Large's model architecture is described below.

4.1.1. Model Architecture.

The bigger version of BERT, also referred to as BERT-Large, consists of a Transformer with 24 hidden layers, i.e. Transformer blocks; a hidden layer size of 1024 nodes; a filter size four times the size of the hidden layer size, i.e. a filter size of 4096; and 16 self-attention heads.

4.2. Pre-training Stage

The pre-training stage is a form of unsupervised learning where the BERT model can be trained on an infinite amount of data and is composed of two sub-processes; the sub-processes consist of Masked LM (MLM) and Next Sentence Prediction (NSP) tasks. Due to the unlimited amount of

data the pre-training stage could be trained on, Google's research team completed this portion of the training using a Wikipedia + BookCorpus corpus; the model's checkpoint and other useful code can be accessed at Google's research GitHub [2].

4.2.1. Masked LM (MLM).

The MLM step of the pre-training stage involves masking 15% of tokenized words in an input sequence and using the embedded matrices of the words surrounding the masked word, BERT was tasked to predict the word being masked. During masking, 80% of masked words were replaced with a "[MASK]" token; 10% with another randomly selected word; and the remaining 10% of the time, the masked word was left unedited. The purpose of MLM was for the model to learn the context of words, since it would be tasked to predict the masked word using the context of the masked word.

4.2.2 Next Sentence Prediction (NSP).

Given two input sentences, during the NSP task, BERT is tasked with predicting whether or not the second sentence occurs directly after the first sentence; the NSP task's end goal is to train the BERT model to relate an individual input sequence's embedded matrix with another sequence's input sequence and in turn group similar sequences together. We will modify NSP in the fine-tuning stage of the BERT model, so it can output the probabilities of a given input sequence being positive or negative.

4.3. Fine-tuning Stage

BERT's pre-training stage makes the model able to be used for a wide range of NLP tasks; however, in the fine-tuning stage, BERT is trained on the specific NLP task the user has in mind. For the IMDB Sentiment Analysis project, BERT's fine-tuning stage will be modified and trained to output the probability that a given input sequence can be classified as positive or negative. For the most part, the training process is the same for the pre-training and fine-tuning stages; however, slight variations have to be applied to the fine-tuning stage.

4.3.1. Training Variation.

Fine-tuning Training Process.

The three main differences between training process of the pre-training and fine-tuning stages are the batch size, number of epochs and learning rate of the Adam optimizer. During fine-tuning, BERT will be trained on a batch size of 32 for 3 epochs. Further, the learning rate of the Adam optimizer is lowered to a value of 2e-5.

NSP Modification.

During pre-training, the NSP task used a binary label to dictate whether or not the model predicted the relationship between sentences correctly; either a False for "NotNextSentence" or a True for "IsNextSentence". During the fine-tuning process of sentiment analysis, the binary next sentence labels will be replaced by a False, or "0", for a review with negative sentiment or True, or "1", for a review with positive sentiment; the second sentence input of NSP can be replaced by a null, or "None", value.

4.4. Model Performance

After pre-training and fine-tuning are both complete, the BERT model is ready to be evaluated; to measure performance, the percentage of correctly predicted movie reviews will be used, i.e. accuracy metric.

4.4.1 Evaluation.

The final trained BERT model was able to achieve a training accuracy of 99.55%; this is very high and could be a sign that the model overfitted, i.e. memorized, the training data; overfitting is a one of the most common issues with neural networks, because it means that the model essentially memorized the training dataset at the cost of being able to correctly classify unseen data, such as the testing dataset. The only way to find out if the model overfitted is to use the model to classify unseen data which will be done in the next section; anything less than 80% is a good indication that the model overfitted to the training dataset.

4.4.2. Prediction.

After classifying the unseen test dataset, BERT achieved an accuracy of about 94.93%; this accuracy translates to BERT correctly classifying the sentiment of 23,732 out of the 25,000 previously unseen movie reviews. The final performance observed surpasses the original goal of 85% accuracy by almost 10%, and its 94.93% accuracy breaks into the current top 10 state-of-the-art scores achieved on the IMDB Sentiment Analysis task.

References

[1] Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142-150.

[2] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *arXiv preprint, arXiv:1810.04805*

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.