COMP 551 - Mini Project 3

Xin Yi Du, Cailean Oikawa, Yanying Zhang

April 13, 2023

1 Abstract

In this project, we implemented two machine learning models: Naive Bayes and BERT on textual data and compared their performance on the IMDB review dataset. From the testing result, we could conclude that BERT model performed a better accuracy than Naive Bayes model, which is consistent with our expectations since BERT is a more advanced algorithm. Through the experiment, we obtained a higher accuracy of 0.92 over 0.90. Thus was able to classify correctly positive and negative reviews. On the other hand, we also implemented Laplace smoothing and compared the two methods, which turned out that the model trained with Laplace smoothing performed a better result than the model without Laplace transformation. And we also considered bagging the Naive Bayes models to reduce variance and this gave us the best results we were able to obtain.

2 Introduction

The task was to take the online movie reviews dataset and train it on Naive Bayes and BERT with pre-trained weights to perform sentiment classification. Several optimization methods were used to improve the accuracy of the Naive Bayes Model such as bagging with random hill climb search, Laplace smoothing, word frequency threshold, etc.

According to the authors of "Learning Word Vectors for Sentiment Analysis", their model "performed best when concatenated with bag of words representation." Our analysis confirmed this finding as both our Naïve Bayes and BERT models were able to classify the movie reviews with over 90% accuracy correctly.

3 Dataset

The dataset for this project is the IMDB movie review dataset which is a common dataset for evaluating sentiment and stance classification models. The IMDB dataset was separated into the training data and the test data comprised of 25,000 movie reviews each, evenly labelled as positive and negative. The dataset also contained 50,000 unlabeled reviews with polarized and neutral sentiment. The unlabeled portion was not used for training or testing. The preprocessing done for the Naive Bayes and BERT models is very different, the BERT model has a specific tokenizer which converts each review into a 512 integer vector which the model accepts as input. The Naive Bayes model's behaviour is defined by the preprocessor so each model we tested used a different combination of preprocessing techniques. We can consider this a form of feature engineering because the multinomial Bayes model we use will behave better or worse depending on the way that each review is tokenized. We employed combinations of the following techniques and selected the best combination via random hill climb.

- Ngram tokens - split review into its word ngrams
- Prefix only - consider only the first n words of the review
- Suffix only - consider only the final n words of the review
- Frequency threshold - omit any low-frequency tokens
- Length threshold - omit any short words from the review

4 Results

We measured the Naive Bayes and BERT models primarily by accuracy. The best Naive Bayes model achieved 90.2% accuracy and the BERT model achieved 92.8% accuracy when trained for 2 epochs. We found that the Naive Bayes model had roughly equivalent numbers of false negatives and false positives, 1274, and 1477 respectively on the testing dataset which had 12500 examples of each label. It is interesting that the BERT model performs slightly better but had a considerable imbalance in the number of false negatives and false positives. BERT gave 494 false negatives and 1289 false positives.

We spent more time optimizing the Naive Bayes models than BERT because they required no time on a GPU and iterative experimenting was much easier with the simpler model. We present here some observations about these models. All experiments use multinomial Bag-of-words Naive Bayes models. The first model experimented with did very minimal processing of the reviews, taking all words in each review and converting to lowercase before computing the multinomial parameters. This model obtains an accuracy of 0.78. Adding Laplace smoothing to this model brings the accuracy to 0.82. The next class of models experimented on were models which considered different n-grams in the reviews as individual tokens. Using 2-grams instead of single words brings the accuracy to 0.87. We observed that different n-gram models have very little overlap in incorrectly classified reviews - we decided to experiment with bagging of multiple Naive Bayes models trained on different subsets of features. To find the best model, we came up with 18 reasonable methods of processing the review data and had a random hill climb algorithm search for the best bag model of 4 out of the 18 proposed models. This improved the accuracy over any single proposed model, illustrating the power of bagging and indicating that our models may have suffered from high variance.

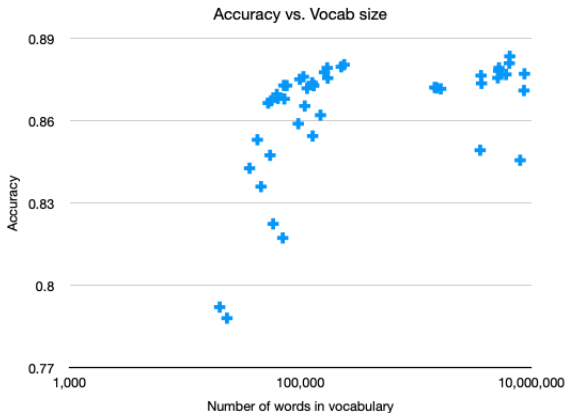
The best Naive Bayes model predicted class labels by taking the average of the following 4 models.

- model 1 trained on the 25 ending words of the review excluding any words with fewer than 5 characters.
- model 2 trained on 3-grams from the entire review.
- model 3 trained on the union of single words and 3-grams in the 50 ending words of the review.
- model 4 trained on the union of 2-grams and 4-grams in the entire review.

All models used Laplace smoothing and considered only words which appeared at least twice in the training set. Below, we discuss the effect of limiting minimum word frequency as a regularization strategy.

It is interesting to note that in the 18 proposed model, there were models which examined only review prefixes and models which examined only words of high frequency but the random hill climb always ruled these out which tells us that the ending of the review contains more information for a naive Bayes model than the beginning.

We performed another experiment where we attempted to regularize various ngram models by filtering out words which are very low frequency or short in length. We used models with a mix of 2-grams,3-grams or 4-grams and limited word frequency to either 1,5, or 10 and limited word length to either 1,2, or 3. We plotted the accuracy score vs. the size of the vocabulary.



We reported 90.2% as the accuracy for the best model which was a bag of 4 Naive Bayes models which considered in total 1305704 different tokens but as we can see it is possible to achieve accuracy of up to 88.0% using a model with only 238989 different tokens. So we consider this method of shrinking vocabulary size an effective regularization strategy which increases bias by a little bit but drastically simplifies the model and speeds up prediction by an order of magnitude.

We can obviously improve upon the Naive Bayes result with more time spent engineering features and using more advanced search algorithms than hill climb. Likewise, the BERT model should be trained for longer to obtain the best possible accuracy.

4.1 Naive Bayes vs BERT

We were very impressed by the power of the simple Naive Bayes models - they are competitive with BERT which is one of the state-of-the-art transformers for natural language tasks. Sentiment classification is a task which clearly does not utilize the full potential of multi-headed attention architecture and for this reason simple n-gram statistics are a pretty good signal of class label. If the task had been stance classification we would expect fairly poor performance with a Naive Bayes model while in theory, the transformer architecture of BERT should perform well with appropriate fine tuning.

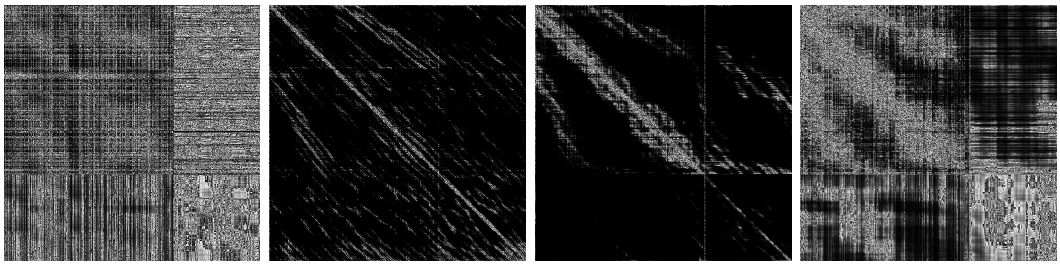
4.2 Attention Matrix

In this experiment, we trained a BERT model configured for attention output for 2 epochs (achieving 92.8% accuracy) and then chose 2 illustrative examples to examine attention matrices for. BERT has 12 times 12 equals 144 different attention mechanisms and each can be examined as a 512 by 512 matrix since each mechanism is a self-attention layer on the 512 tokens BERT accepts as input. The first instance we examined is the very first review in the test set which BERT classifies correctly as negative.

The review starts something like this

```
'[CLS]', 'i', 'love', 'sci', '-', 'fi', 'and', 'am',
'willing','to', 'put', 'up', 'with', 'a', 'lot', '.'
```

We include here some of the attention matrices extracted from the models forward pass on review 0 represented as 512 by 512 grayscale bitmaps.

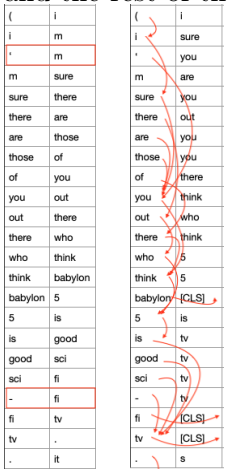


The above attention matrices are indexed (0,1),(1,1),(5,3),(8,1) in the multiheaded attention layer these models were selected because they visually show that certain attention heads are learning local information while others are learning longer range patterns.

We consider the tokens 107-135 from review 0.

```
'(', 'i', "'", 'm', 'sure', 'there', 'are', 'those', 'of', 'you','out', 'there', 'who',
'think', 'babylon', '5', 'is', 'good','sci',
'-', 'fi', 'tv'
```

We show for two different attention matrices the maximum connection between the tokens in this section and the rest of the review.



(	i
i	m
'	m
m	sure
sure	there
there	are
are	those
those	of
of	you
you	out
out	there
there	who
who	think
think	babylon
babylon	5
5	is
is	good
good	sci
sci	fi
-	fi
fi	tv
tv	.
.	it

(	i
i	sure
'	you
m	are
sure	you
there	out
are	you
those	you
of	there
you	think
out	who
there	think
who	5
think	5
babylon	[CLS]
5	is
is	tv
good	tv
sci	tv
-	tv
fi	[CLS]
tv	[CLS]
.	s

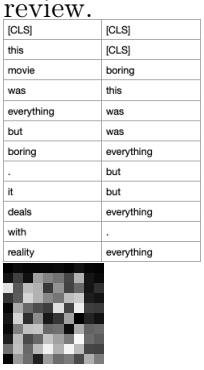
Notice that the first example appears to be attending to all next tokens except when the next token is an irrelevant punctuation such as the apostrophe in "i'm" and the hyphen in "sci-fi" which are skipped by this attention matrix. So this could be an instance of an attention matrix ruling out some irrelevant tokens for the current task. The second is an attention matrix which appears to be looking ahead to the following highly relevant word. Tokens map very strongly to the tokens "you" and "tv" which are highly relevant to the semantics of the sentence.

We will now consider a different review. Review 12541 in the test dataset is labelled positive but the BERT model classifies it as negative.

The review starts off

```
'[CLS]', 'this', 'movie', 'was', 'everything','but',
'boring','.', 'it', 'deals', 'with','reality'
```

We can examine the attention matrices to get some idea of why the model is incorrectly classifying this review.



[CLS]	[CLS]
this	[CLS]
movie	boring
was	this
everything	was
but	was
boring	everything
.	but
it	but
deals	everything
with	.
reality	everything

Notice that this particular attention matrix appears to associate the token "movie" with "boring". This is only one attention matrix of many and it does not entirely define the model's behaviour but we can speculate that the double negative in the review may have tricked the model into associating a positive review with negative sentiment. We can also see by examining the 10 by 10 section of the full attention matrix that there is a lot of entropy and it is possible that more training would have resulted in correct classification.

4.3 Pre-training

The BERT model which stands for Bidirectional Encoder Representations from Transformers was pre-trained by Google in 2018 and remains a powerful model today for NLP tasks. BERT was trained on two tasks, **Language Modeling**, and **Next Sentence Prediction**. This pre-training was incredibly computation intensive and produces a model with a powerful ability to perform generic NLP tasks which exploit it's latent representation of language via some low computation intense fine tuning.

4.4 Deep Learning vs Traditional ML

It is worth noting that the difference between Naive Bayes and BERT is not very severe, we could have trained the BERT model for longer in order to obtain a higher result but the Naive Bayes model being able to achieve over 90% accuracy is a good result considering the very sweeping assumptions it makes about conditional independence. We could conclude that sentiment classification is not a task so advanced that it necessarily requires state-of-the-art transformer models but can effectively be done via traditional ML techniques.

5 Conclusion

As a result, we could conclude that BERT performed better result than Naive Bayes algorithm. Besides the advanced result shown in the accuracy aspect, the Naive Bayes model with Laplace smoothing also works out with better accuracy than without the technique of Laplace smoothing. Due to the zero probability issue, Laplace smoothing perfectly handle the case and increase the accuracy rate by reducing the zero probability model. Meanwhile, since the dataset is highly dispersed, we explore through hill climbing method to deal with the problem of multiple local mins. This turns out to be a better result than the plain Naive Bayes model. But in another word, we could also try other methods such as combining the greedy algorithm with hill climbing algorithms.

6 Statement of Contributions

All group members contributed fairly and equally to this project.

code-Cailean Oikawa report-Xin Yi Du, Yanying Zhang