Bag of N-Gram Document Classification

Shubham Chandel - sc7238@nyu.edu

October 10, 2018

1 Introduction

Document classification is a classification problem in Machine Learning, which has been approached in various ways in the past. In this work, we aim to evaluate how well a Bag of N-Gram model will work on this type of classification problem.

More specifically, we use the IMDB Movie review dataset [1], to perform sentiment analysis, which is a binary classification problem.

A comprehensive ablation study is conducted and results are presented in the report and code is available here. https://github.com/sksq96/nyu-nlp

2 Bag of N-Gram Classification

The Bag of Words [2] model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears.

In this work, we extend this to Bag-of-N-gram for predicting the sentiment of the movie reviews given the textual description for by the user.

3 Experiments

3.1 Dataset

We make use of IMDB Large Movie Review Dataset, which is a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. Each review is provided with a binary label indicating the sentiment of the review.

3.2 Experimental set-up

To evaluate the effect of different hyperparameters on the performance of our model, we fix an initial set of parameters presented in Table 1.

n-gram	vocabulary size	embedding size	max sentence length	optimizer	learning rate	tokenization
2	10,000	100	200	Adam	1e-3	HTML Tags

Table 1: Initial set of Parameters

Once we have these set of parameters, we change one of these, keeping all other constant. The motivation behind this is to analyze what effect each of them have on the final model.

We split the train dataset into 20,000 train examples and 5,000 validation examples.

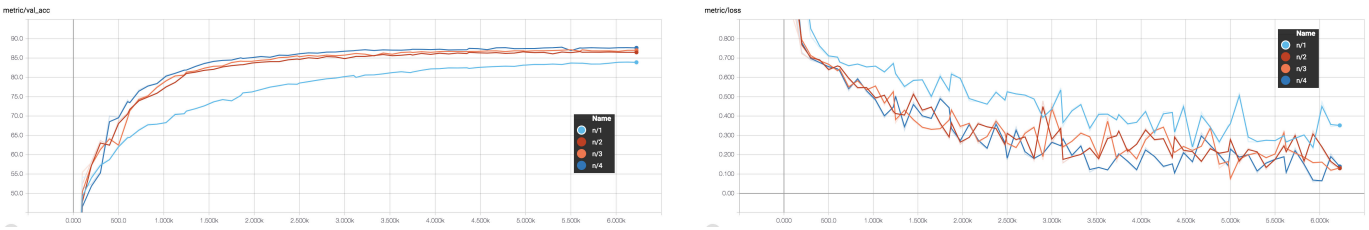
4 Results and interpretation

In this section, we look at detailed ablation study for each hyperparameter evaluated and a brief discussion surrounding each is presented to the readers.

4.1 N Gram

The most notable difference is prominently seen when moving from the 1-gram model to a 2-gram model. The 1-gram model has a slower convergence and the final accuracy achieve is about 4% less than other gram models as shown in Table 2.

What's even more interesting is, we observe very slight performance gap, moving from n=2 to n=3 and n=4. A possible explanation is, moving beyond 2-gram, the frequency of n-gram keywords to existing in top-k is drastically reduced.



(a) (b)

Figure 1: Validation accuracy as a function of timestep for different N-Grams.

n=1	n=2	n=3	n=4
83.89%	86.48%	87.07%	87.63%

Table 2: Validation accuracy for different N-Grams.

4.2 Optimizer

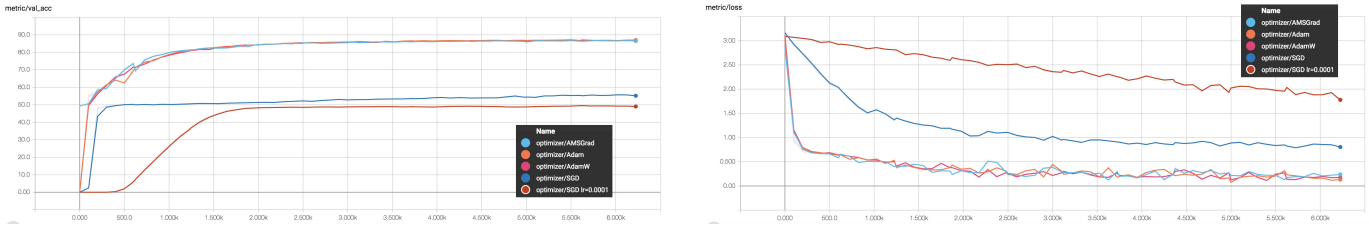
One of the striking observation is how reliable is Adam [3] out of the box. We tested various optimization schemes and Adam performed the best, even surpassing the latest AMSGrad [4] and even AdamW [5].

AdamW claims to provide super-convergence and is expected to solve the problems posed by Adam and AMSGrad. However, in our experiments, the results are on the contrary. Neither were we able to achieve super-convergence, now did AdamW resulted in a lower loss than it's counterparts.

On the other hand, SGD [6], shown in Table 3, proved to be difficult to tune and resulted in poor results at best. A slower SDG with a learning rate of 0.0001, is optimized slowly as one would expect it to be. Alas, SGD fails to obtains decent results, resulting in just marginal better result than a coin-flip.

Adam	AMSGrad	AdamW	SGD lr=0.001	SGD lr=0.0001
87.02%	86.56%	86.52%	55.13%	49.02%

Table 3: Validation accuracy for different Optimizers.



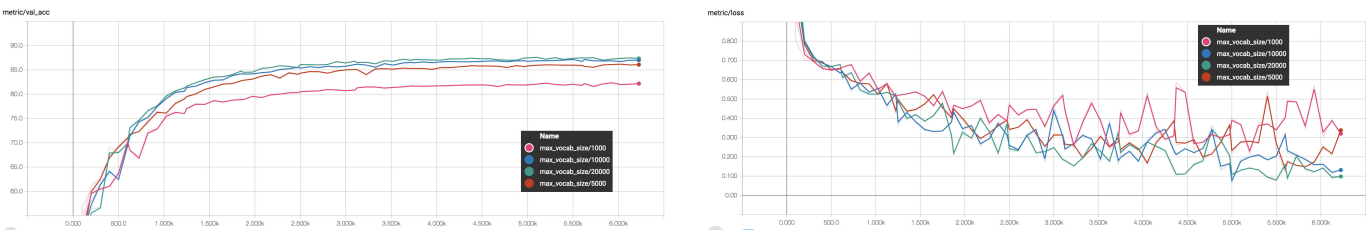
(a) (b)

Figure 2: Validation accuracy as a function of timestep for different Optimizers.

4.3 Vocabulary Size

We observe, a limited vocabulary of 1000 is limited by it's validation performance. Further, vocabulary sizes of 5000, 10000 and even 20000 perform nearly well on the validation dataset, as indicated in Table 4.

An average young human has a vocabulary of 10000, so the model with access to these many tokens performs reasonably well on this relatively easy problem at hand.



(a) (b)

Figure 3: Validation accuracy as a function of timestep for different Vocabulary Size.

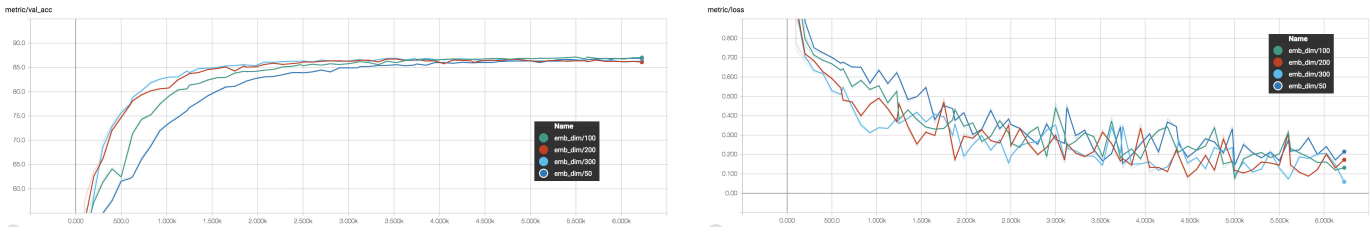
20000	10000	5000	1000
87.62%	87.34%	86.08%	82.18%

Table 4: Validation accuracy for different Vocabulary Size.

4.4 Embedding dimension

We observe less than 1% validation accuracy drop going from embedding dimension of n=50 to n=300.

We hypothesize, the task at hand and the naive bag-of-n-grams model, do not require us to store complex embeddings so, even a small embedding size of 50 provides competitive performance as provided by the standard 300 size one.



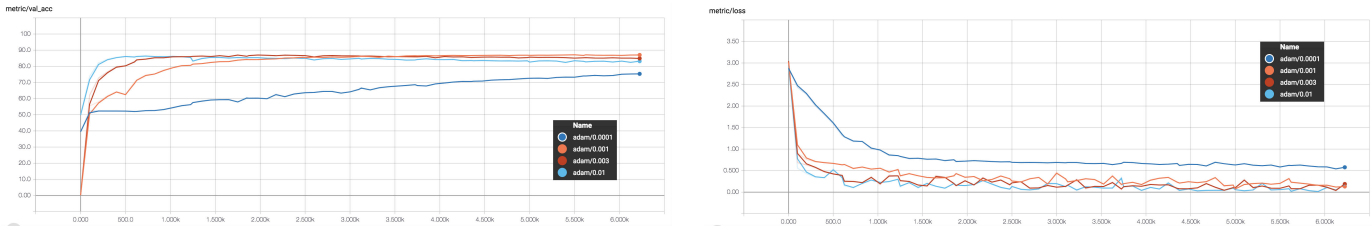
(a) (b)

Figure 4: Validation accuracy as a function of timestep for different Embedding dimensions.

4.5 Learning Rate

We experiment evaluated learning rate ranging from 1e-4 to 1e-2. We observe 1e-4 to perform the worst, both in terms of the convergence rate and the final accuracy achieved. All other learning rates result in similar validation accuracy.

This observation strengthens the notion of robustness of the Adam optimizer to various learning rates, in contrast to SGD which requires carefully fine-tuned learning rate.

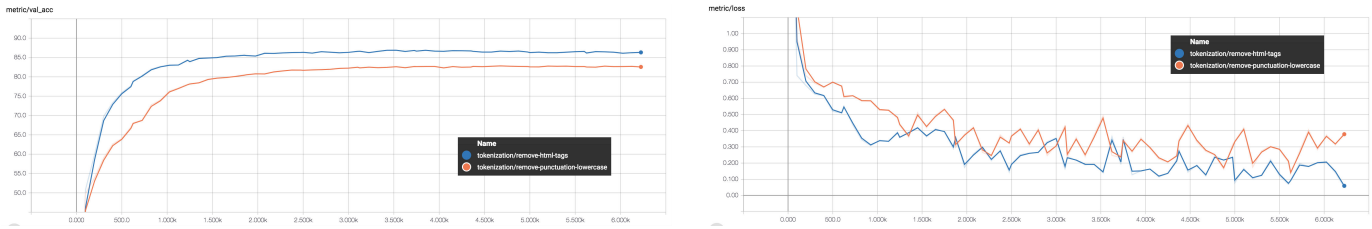


(a) (b)

Figure 5: Validation accuracy as a function of timestep for different Learning Rates.

4.6 Tokenization

The data distribution for IMDB reviews comes from a webpage. The tokenization scheme to remove the HTML tags works decently well as shown in Figure 6.

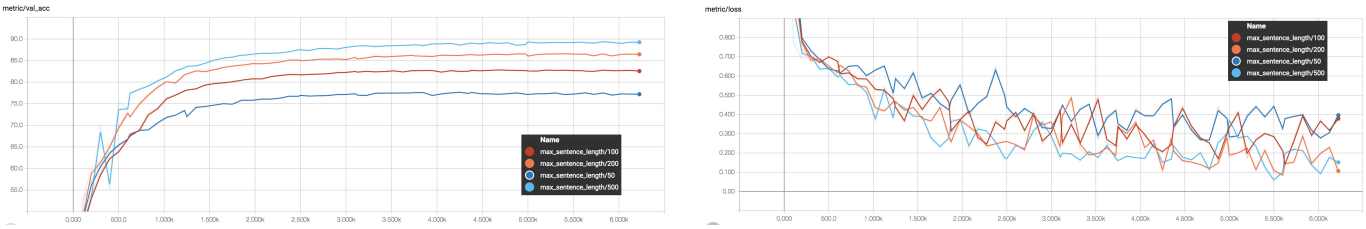


(a) (b)

Figure 6: Validation accuracy as a function of timestep for different Tokenization.

4.7 Maximum Sentence Length

In our experiments, this parameter had a direct effect on the final loss and by virtue of it, on accuracy. We observe more than 12% boost in accuracy, as indicated in Table 5, by increasing maximum length from 1 to 4.



(a) (b)

Figure 7: Validation accuracy as a function of timestep for different Maximum Sentence Lengths.

This observation is due to increasing model complexity. Increasing maximum input sentence length to the model will result in training a bigger network which will, in turn, be responsible for a lower loss.

500	200	100	50
89.24%	86.44%	82.54%	77.20%

Table 5: Validation accuracy for different Maximum Sentence Lengths.

5 Incorrect and Correct predictions

5.1 Incorrect predictions

1. Predicted: 0, True: 1

   In my opinion, this film has wonderful lighting and even better photography. Too bad the story is not all that good and Mr. Cage sometimes loses his accent. But two thumbs up for lighting and the DP!

   Analysis: The bag-of-words model likely picked up on "too bad" and it the sentiment of the user is not clear.

2. Predicted: 0, True: 1

   I don't care if some people voted this movie to be bad. If you want the Truth this is a Very Good Movie! It has every thing a movie should have. You really should Get this one.

   Analysis: Again the model picked up on "to be bad" and this is weakness of this naive classifier.

3. Predicted: 1, True: 0

   My first thoughts on this film were of using science fiction as a bad way to show naked women, althought not a brilliant story line it had quite a good ending

   Analysis: The tokens "bad", "not a" indication to model for a negative prediction. Again, the intent of author not clear.

5.2 Correct predictions

1. Predicted: 0, True: 0

   I can't believe this movie has an average rating of 7.0! It is a fiendishly bad movie, and I saw it when it was fairly new, and I was in the age group that is supposed to like it!

   Analysis: The text presents strong keywords like "fiendishly" and "bad movie" leading to correct prediction.

2. Predicted: 1, True: 1

   Wonderful movie. Adult content. Lots of erotic scenes plus excellent music and dance scenes. My wife and I absolutely loved this movie and wish they'd make more like it.

   Analysis: A number of positive keywords in the review.

3. Predicted: 0, True: 0

   I thought this movie was horrible. I was bored and had to use all the self control I have to not scream at the screen. Mod Squad was beyond cheesy, beyond cliche, and utterly predictable.

   Analysis: The review has tokens such as, "movie was horrible", "bored", "scream", "utterly predictable" making it easy for this naive classifier.

References

[1] Large Movie Review Dataset http://ai.stanford.edu/∼amaas/data/sentiment

[2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space

[3] Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization

[4] Sashank J. Reddi and Satyen Kale and Sanjiv Kumar. On the Convergence of Adam and Beyond

[5] Sylvain Gugger and Jeremy Howard. AdamW and Super-convergence is now the fastest way to train neural nets · fast.ai

[6] Bottou, Léon. Large-scale machine learning with stochastic gradient descent