Movie Review Sentiment Analysis

Abstract

The purpose of this research is to conduct the sentiment analysis towards the movie reviews using different methods that's commonly used in the NLP world.
Several models will be created with specific features to be tested onto the same movie review dataset.
The models' performances will be evaluated through the accuracy of its predictions on the holdout data after certain rounds of training.

Introduction

Sentiment analysis is my initial learning objective of NLP when I was first introduced to this field. To extract the main idea or attitude from such unstructured data would help a lot in organizing the information and serve as the basis of any further analysis.
Also, the movie review dataset is one of the most classic dataset that has been used for sentiment analysis. Such data should support my purpose here to find an effective text classification model.
A model that could help understand and summarize other people's comments should be a very useful tool, which could be applicable in a lot relevant fields. For instance, analyzing the reviews of books to make recommendations accordingly, or studying the customers' feedback of certain products to better understand the market.

Background

The sentiment analysis has been conducted for many years with different approaches.
From Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, where I got the data from, it suggested to use Recursive Neural Tensor Network which is something similar as LSTM to me that read phrases to help predict sentences rather than single words.
CNN is another major trends from literature to perform text classification. This methodology is relatively easy to conduct and explain. Furthermore, its strong performance in this field also indicate it to be a good candidate for this particular research.
Last but not least, aspect-based sentiment analysis is showing strong predictive power among all the other BERT models. It identifies fine-grained polarity towards a specific aspect associated with a given target. Although during this research, with time and computing power constraints, no BERT-ASBA model was evaluated; the results from all experiments were discussed in the end to compare with this advanced methodology.

Method

Data

The data is taken directly from a GLUE task - The Stanford Sentiment Treebank.
The data is already cleaned for the analysis; therefore, minimum data pre-processing is required for the research.
The only adjustment made from the raw input to this analysis is to combine the split training and validation datasets and ignoring the test dataset which is missing the label as the nature of a GLUE task. There are still 68,221 records remaining for this research which should be sufficient.
More specifically, there are 67,349 records in the existing training file, and 872 records from the validation file. Therefore, with an 80-20 split for this research's modeling purpose, I will have 54,577 records for training and 13,644 records for testing.
Before start modelling, some explanatory analyses were conducted to ensure the data is good for use.

After randomly split the combined data into training and holdout, the balance of the label is also checked. As 55% versus 45% of positive versus negative distribution is relatively comparable, no further treatment was made towards the data.

Baseline

The baseline analysis for this research is the most basic version of the experiments.
Deep Averaging Network should be the simplest structure to begin with.
By comparing the valuation accuracy from deep averaging network with
(1) “word2vec” embedding initializer, without embedding retraining
(2) “word2vec” embedding initializer, with embedding retraining
(3) “uniform” embedding initializer, with embedding retraining
The first option gave me the worst result when three models are trained by the same parameters. And then followed by the third option, which is actually not much worse than the second option after ten epochs.
Therefore, it seems that the initial embedding plays an important role but not as effectiveness as the retraining.
For the further experiments, to improve the valuation accuracy, I will need to make sure the embedding is retrained. Besides, starting from a pre-trained embedding should also help with the performance.

Experiments

There are several experiments being conducted here from basic to advanced level to compare and conclude which method and feature could provide higher accuracy of the predictions on the testing dataset.

Experiment 1

Deep Averaging Network vs Weighted Averaging Network
With all the parameters set to be the same for both models, the only difference between the weighted averaging model is using attention to weight the input.
As expected, with the averaging structure change, the accuracy improved in Weighted Averaging Network after the same epochs of training.
Therefore, attention layer, or weights towards the input token, should be considered in the following experiments if possible to achieve better result.

Experiment 2

BERT cased vs BERT uncased
The results from models using pre-trained BERT embedding are pretty close for cased versus uncased. Generally speaking for the runs after eight epochs, cased perform slightly better than the uncased in this experiment.
With the consideration of movie comments’ nature that there should be a lot names included. Case sensitive embedding makes more sense to help differentiate a word’s part of speech, and therefore to implicate for its real meaning.
In conclusion, for the following experiments, cased BERT will be used as the initial embedding.

Experiment 3

pooled token vs simple average
The comparison between these two models is relative clear comparing to the second experiment.
Simply by comparing the accuracy and losses from the valuation dataset, it was obvious that the pooled token model provided more desirable result.
Another thing to notice is that the training accuracy increased dramatically through epochs from the model taking simple average of BERT output, while its valuation’s accuracy did not

show increase as steep as the training did, which potentially indicated overfitting in the model.

Experiment 4

pooled token vs CLS token

While it was not that obvious from the second experiments, there could be some overfitting issue; after the third experiment, parameters should be considered to tweak to fit the model.
When comparing the valuation dataset's performance, it is noticed that especially for the model with CLS token, although the accuracy was increasing over epochs, the losses on valuation data were also increasing through the history.
Even though no clear distinction in valuation dataset's accuracy has been identified between these two token methods, by comparing the valuation losses from the history, model using CLS token showed a little worse performance than the model using pooled token.
From the modelling history, other than the valuation's losses, it was observed that there was a decreasing trend in training's losses and an increasing trend in training's accuracy, which were as expected. In the meanwhile, the valuation dataset's accuracy showed a relative stable trend over epochs, which was concerning.
Such undesirable trends observed from valuation dataset for both losses and accuracy should serve as an evidence of overfitting to the training data.
With the consideration of model complexity, a larger learning rate is applied to these two models to prevent overfitting.
During the re-run of this experiment, model using CLS token performed better than the model with pooled token a little bit in terms of valuation dataset's accuracy, which is even outperforming than its own training dataset's accuracy a lot. Although the losses still not showing any decreasing trend over epochs from valuation dataset, at least no obvious increasing trend could be observed either. Such change should be able to justify as an improvement from the last run.
The similar improvements were observed from both models although the model using pooled token showed much smaller gap between training versus valuation datasets in terms of the accuracy.
The results from these two rounds of experiments implied that the learning rate used for training should match with the model's complexity to provide more reasonable results.
Based on this experiment, both pooled and CLS tokens are performing better than the simple average output, which suggested advanced treatment on top of the BERT output is preferred to have more accurate predictions.

Experiment 5

BERT + CNN vs BERT + LSTM

To prevent the overfitting that was observed from the previous experiments, an early stopping mechanism was introduced during this experiment. With the consideration of added complexity of either CNN or LSTM, the learning rate kept as small as 0.00005 in this experiment.
The working logic of early stopping is when it identifies after certain number of epochs, the specified field is not improving, which is the valuation accuracy in this experiment, the model will automatically stop running.
Based on the accuracy result from these two models, early stopping was not triggered in BERT with CNN model but helped to terminate BERT with LSTM model after six epochs.
Also by comparing both losses and accuracy from the valuation data, BERT with CNN on top always performed better than the model with LSTM added on top.
Another interesting finding from this experiment is that the losses' trends were pretty similar as the rerun from the fourth experiment. Assuming the overfitting issue was resolved when the learning rate was increased from last experiment, the much smaller learning rate used in this experiment became a good fit to such advanced architecture.

Results and Discussion

Model	Features	Valuation Loss	Valuation Accuracy
Deep Averaging Network	word2vec embedding, not retrain embedding	0.3833	0.8240
Weighted Averaging Network	word2vec embedding, not retrain embedding	0.3417	0.8481
Deep Averaging Network	uniform embedding, retrain embedding	0.4270	0.8838
Deep Averaging Network	word2vec embedding, retrain embedding	0.4906	0.8819
Weighted Averaging Network	word2vec embedding, retrain embedding	0.5389	0.8854
BERT	uncased embedding, pooled token	0.2971	0.9305
BERT	cased embedding, pooled token	0.2437	0.9352
BERT	cased embedding, CLS token	0.3506	0.9307
BERT	cased embedding, average output	0.6163	0.9111
BERT + CNN	cased embedding	0.3305	0.9325
BERT + LSTM	cased embedding	0.6599	0.9256

Although the differences between several adjacent experiments were not easily distinguishable with the consideration of both loss and accuracy from the valuation data; when treating the whole research into several buckets, it became much easier to differentiate between different stages.

The first stage was the averaging network without retraining the embedding. Although having the attention layer improved the results, it still cannot compare to the model without attention but retraining the embedding.

Following the first stage, having retrained embedding led the experiments jumped to a higher level in terms of accuracy of the valuation data. Although at the same time, the losses from valuation data were affected which were higher than the first stage.

The introduction of BERT was totally a game changer in this research, which ended to our last stage. From here, the accuracy from valuation data were consistently higher than all previous models. However, on the same time, after comparing different ways of summarizing information from the BERT output, adding advanced layers on top of BERT was no longer showing any apparent further enhancement to the model.

Although from the research literature, CNN should be very helpful for text classification, the result from the experiment was not presenting any convincing evidence.

Due to the time constraint, no successful BERT-ABSA model was performed onto this dataset. Based on the literature research, the highest accuracy by performing aspect-based sentiment analysis could reach as high as 0.94. With the consideration of data quality, the performance is expected to be somehow higher than the research paper if the model was successfully run on this movie review dataset.

But from another perspective of view, there might not be too much room left for the model to improve on this dataset - which might be an explanation of why the BERT with CNN model did not stand out from this research.

Conclusion

The most important takeaway I learned from this research is that the pre-trained embedding should be well utilized when performing any NLP related tasks.

The experiment with BERT alone being the winning model of this research indicated:

(1) pre-trained embedding is super helpful.

(2) for a simple datasest, advanced architecture might not add too much value.

From the experiments, it was also noticed that the learning rate and the complexity of the architecture should be positively correlated; otherwise, there could be overfitting issues

Reference

Sentiment Analysis <https://nlp.stanford.edu/sentiment/index.html>
A Sensitivity Analysis of Convolutional Neural Networks for Sentence Classification <https://arxiv.org/abs/1510.03820>
Convolutional Neural Networks for Sentence Classification <https://arxiv.org/abs/1408.5882>
Understanding LSTM Networks <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
Recurrent Neural Network Based Language Model <http://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf>
Context-Guided BERT for Targeted Aspect-Based Sentiment Analysis <https://www.semanticscholar.org/paper/Attention-Enhanced-Graph-Convolutional-Networks-for-Xu-Liu/bf6895dda201fd3596197952f2191dc597cd8346>
SA-ASBA: a hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting <https://www.researchgate.net/publication/364679271_SA-ASBA_a_hybrid_model_for_aspect-based_sentiment_analysis_using_synthetic_attention_in_pre-trained_language_BERT_model_with_extreme_gradient_boosting>
BERT for ABSA <https://github.com/LorenzoAgnolucci/BERT_for_ABSA>