

Natural Language Processing

Movie Reviews Classification using DistilBERT

Marina Samprovalaki *f3322310*

Load the Dataset

In this study, we utilized the [Hugging Face's 'datasets' library](#) to load the [IMDb dataset](#), a widely used benchmark for sentiment analysis. The dataset consists of movie reviews categorized into positive and negative sentiments. Due to the large size of the IMDb dataset, we opted to work with smaller samples for both the training and test sets. This strategy enables us to create manageable yet representative subsets, facilitating efficient experimentation and analysis in the context of sentiment analysis models.

Tokenization

We instantiated a sequence classification model, specifically the ['distilbert-base-uncased' model](#). The model is designed for binary classification tasks with two labels. This pre-trained model is ready for fine-tuning on the tokenized IMDb dataset for sentiment analysis.

We also created a data collator, which is configured with the provided tokenizer and is intended for handling padding during batched processing of tokenized data. It plays a crucial role in preparing the input data for training the sentiment analysis model on the IMDb dataset.

Dataset Subdivision


Here are the sizes of the employed datasets derived from the initial pool of 50k documents.

Dataset	Size
Training	3000
Testing	300
Development	100

Table 1: Utilized Dataset Sizes


Baseline Models

Regarding the Baseline Models, we implemented a Majority Classifier and the top-performing model from Part 2, which is a Logistic Regression classifier incorporating Lasso Regularization. We also used the MLP we created in Part 3 and the Bi-RNN model from Part 4.




	precision	recall	f1-score	support
0	0.50	1.00	0.67	2621
1	0.00	0.00	0.00	2629
accuracy			0.50	5250
macro avg	0.25	0.50	0.33	5250
weighted avg	0.25	0.50	0.33	5250

Figure 1: Dummy Classification Table



	precision	recall	f1-score	support
0	0.50	0.55	0.53	2621
1	0.51	0.46	0.48	2629
accuracy			0.51	5250
macro avg	0.51	0.51	0.50	5250
weighted avg	0.51	0.51	0.50	5250

Figure 2: Logistic Regression - Lasso Classification Table



165/165 [=====] - 1s 7ms/step				
	precision	recall	f1-score	support
0	0.50	0.11	0.18	2634
1	0.50	0.89	0.64	2616
accuracy			0.50	5250
macro avg	0.50	0.50	0.41	5250
weighted avg	0.50	0.50	0.41	5250

Figure 3: MLP Classification Table

Training Process

For the training phase, we utilized the 'Trainer' class from the 'transformers' library. The training arguments were configured to define essential parameters for the training process, such as output directory, learning rate, batch sizes, number of training epochs, weight decay, and evaluation strategy. The training process was initiated using the `trainer.train()` method, enabling the model to learn and optimize its parameters based on the provided IMDB dataset.

	precision	recall	f1-score	support
0	0.51	0.81	0.63	2621
1	0.55	0.23	0.32	2629
accuracy			0.52	5250
macro avg	0.53	0.52	0.47	5250
weighted avg	0.53	0.52	0.47	5250

Figure 4: Bi-RNN with GRU cells Classification Table

	precision	recall	f1-score	support
0	0.50	0.08	0.13	2621
1	0.50	0.92	0.65	2629
accuracy			0.50	5250
macro avg	0.50	0.50	0.39	5250
weighted avg	0.50	0.50	0.39	5250

Figure 5: CNN Classification Table

Evaluation

To assess the performance of the trained model, we employed the `trainer.evaluate()` method. This step involves evaluating the model on the test dataset, providing insights into its accuracy and effectiveness in sentiment analysis on the IMDB dataset. The evaluation results contribute to understanding the model's generalization and its ability to accurately classify sentiments in unseen data.

```

trainer.evaluate()
[19/19 00:04]
{'eval_loss': 0.3773174583911896,
 'eval_accuracy': 0.8866666666666667,
 'eval_f1': 0.8903225806451613,
 'eval_runtime': 5.448,
 'eval_samples_per_second': 55.066,
 'eval_steps_per_second': 3.488,
 'epoch': 3.0}

```

Figure 6: Evaluation parameters

Hyperparameters Tuning

During the hyperparameter tuning phase, a methodical investigation is performed to optimize the sentiment analysis model by exploring learning rates, batch sizes, weight decays, and epochs. It is essential to acknowledge that the use of a smaller subset in the training, development, and test datasets might result in a validation accuracy of 1. This

limitation is attributed to the reduced dataset size but is designed to efficiently explore hyperparameter configurations for identifying optimal settings in sentiment analysis. The ultimate goal is to deploy the final model, trained with the best hyperparameters, for more extensive applications on larger datasets.

In-Context Learning

We employed ChatGPT 3.5 for in-context learning by providing a prompt and examples to test the model's ability to categorize the sentiment of movie reviews. The prompt given before the examples is as follows:

As an expert in movie criticism, I need your discerning skills to categorize a series of movie reviews. Please determine whether each review reflects a positive or negative sentiment based on your cinematic expertise, considering the nuanced expressions within the context of cinematic critique. I will provide examples to start with, and remember, you can only classify them into two classes: positive or negative.

The following reviews were given to ChatGPT for in-context learning:

1. **Text:** 'It was great to see some of my favorite stars of 30 years ago including John Ritter, Ben Gazzarra and Audrey Hepburn... It ain't no "Paper Moon" and only a very pale version of "What's Up, Doc".'
Sentiment: Negative
2. **Text:** 'If the crew behind "Zombie Chronicles" ever read this, here's some advice guys... Only for zombie completists.'
Sentiment: Negative
3. **Text:** 'I have not seen many low budget films I must admit, but this is the worst movie ever probably... Awful simply awful.'
Sentiment: Negative
4. **Text:** 'I never saw it on TV but rented the DVD through Netflix... I'm very disappointed that it was cancelled. I wish they would produce more episodes. Or perhaps a movie.'
Sentiment: Positive
5. **Text:** 'PERHAPS in an attempt to find another "Hot Property" for adaptation... It may not be as witty as Spielberg's classic TV series, but it's still good.'
Sentiment: Positive

6. **Text:** 'At least for a half hour a week... I hope that it finishes this season off well, and is renewed for future seasons. Otherwise, I may never find a reason to watch the big 3 again.'

Sentiment: Positive

The test instances and the model's outputs are concisely presented in Table 2, alongside the correct labels.

#	Review	ChatGPT Response	True Sentiment
1	'This flick is a waste of time... This is a cheap low type of action cinema.'	Negative	Negative
2	'Well every scene so perfectly presented... All in all a great work of art. A work of a masterman.'	Positive	Positive
3	'This is a classic continuation to Bleu... Brilliant film.'	Positive	Positive
4	'Low budget horror movie... If you enjoy this movie, try Committed from 1988 which is basically a rip off of this movie.'	Negative	Negative
5	'Lowe returns to the nest after, yet another, failed relationship... In short, this movie was so boring, I could not even sleep through it! 1 out of 10 stars!'	Negative	Negative
6	'I believe I share the same psychological outlook on the world with Kieslowski... Brilliant film.'	Positive	Positive

Table 2: Sentiment Analysis for Test Movie Reviews

As observed, ChatGPT correctly classified the sentiment of each review.

Conclusion

In conclusion, Table 3 provides a comprehensive overview of the performance comparison among all the models throughout.

Model Name	Val Accuracy	Test Accuracy
Majority Classifier	49.24%	49.71%
Logistic Regression - L1	88.83%	88.69%
MLP - L1	86.94%	87.10%
Bi-RNNs - Attention Layer	85.62%	85.87%
CNNs	86.89%	86.13%
distilbert-base-uncased	100%¹	86.66%

Table 3: Performance of Different Models

¹small subset of the whole dataset