NLP Text Classification

Introduction

- In this project we defined our task to be a text classification task.
- Amazon dataset was given to apply text classification upon the products known as the laptop gears.

MS1

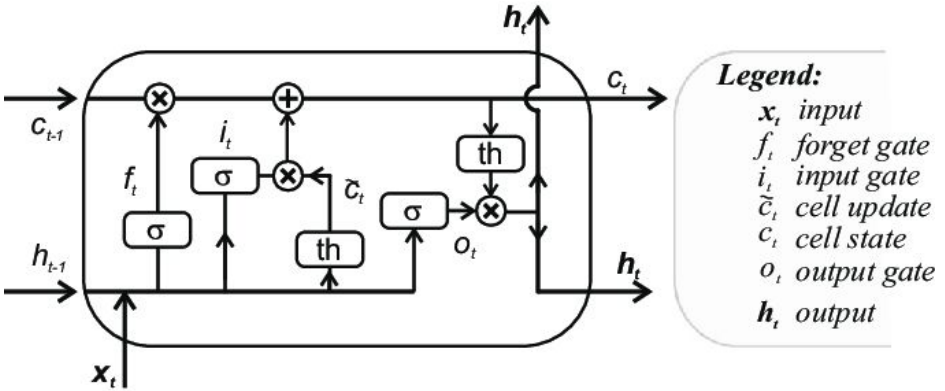
- Exploring the Dataset
- Filling in Empty/ Null Values
- Joining, Cleaning, and Lemmatizing Text Columns
- Sentiment Analysis

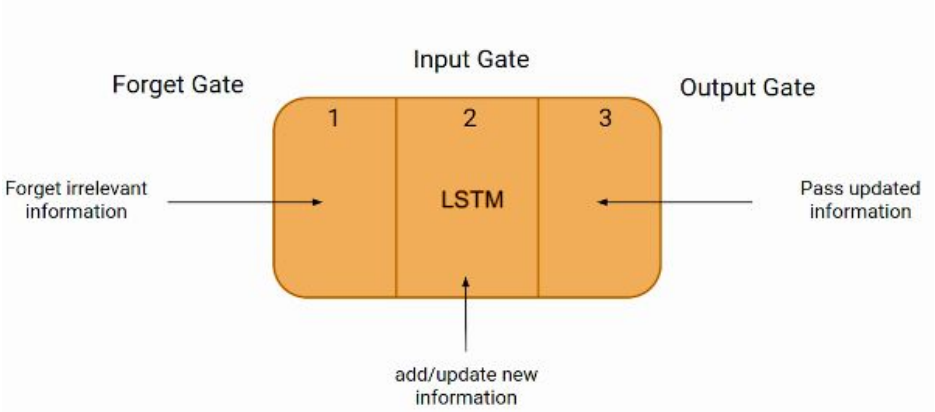
MS2

- Implementing a model from scratch
- Building a Long short term memory (LSTM) recurrent neural network.
- LSTMs developed to capture long term dependencies in Sequential Data

LSTM Model Architecture

- LSTM Cell Structure.
- LSTM Cell Gates.





Preparing the data

- Appending the tokens by the brand name five times to be the prefix and putting it a new column.
- Creating 8 categories on 3 fields of Data as follows:
    - Good Price, Good Sentiment, Good Rating.
    - Good Price, Good Sentiment, Bad Rating.
    - Good Price, Bad Sentiment, Good Rating.
    - Good Price, Bad Sentiment, Bad Rating.
    - Bad Price, Good Sentiment, Good Rating.
    - Bad Price, Good Sentiment, Bad Rating.
    - Bad Price, Bad Sentiment, Good Rating.
    - Bad Price, Bad Sentiment, Bad Rating.

How Categorization done:

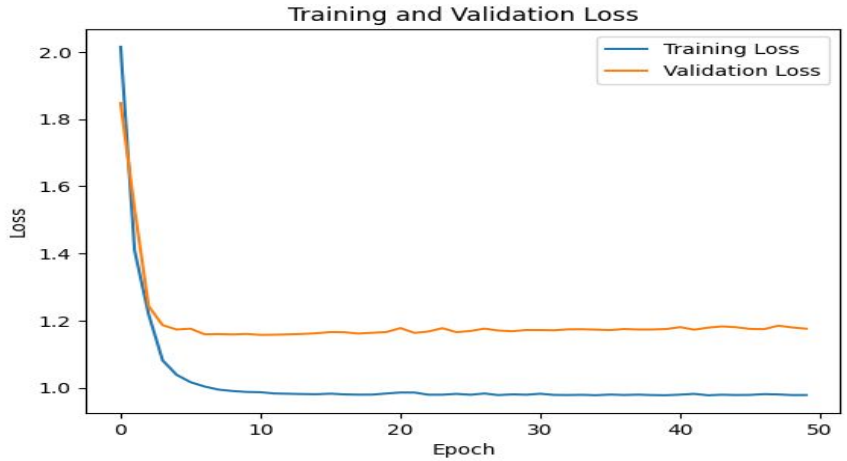
- The price is good when it is less than or equal 50.
- The sentiments labels are good when the sentiment label is positive or neutral.

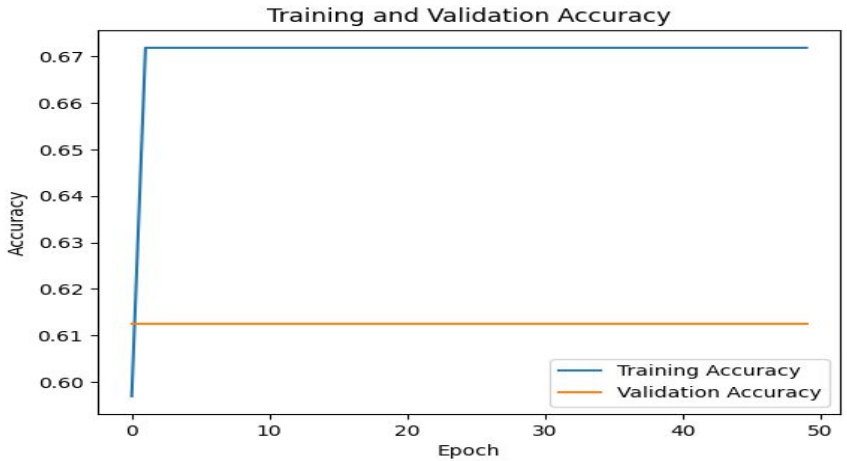
Word2Vec and Neural Embeddings

- A vocab Size of 1000 was chosen by text vectorization to find the unique tokens.
- Text brand column was created to have the data to be trained as a String text.
- The encoded vocab is passed to Word2Vec to get the neural embeddings of each word.

Training and Evaluation

- 50 epochs were trained by splitting the data to 80% training and 20% testing and validation.
- Predictions of the model were printed.
- Testing and Validation Loss and Accuracy of 68% in Testing.

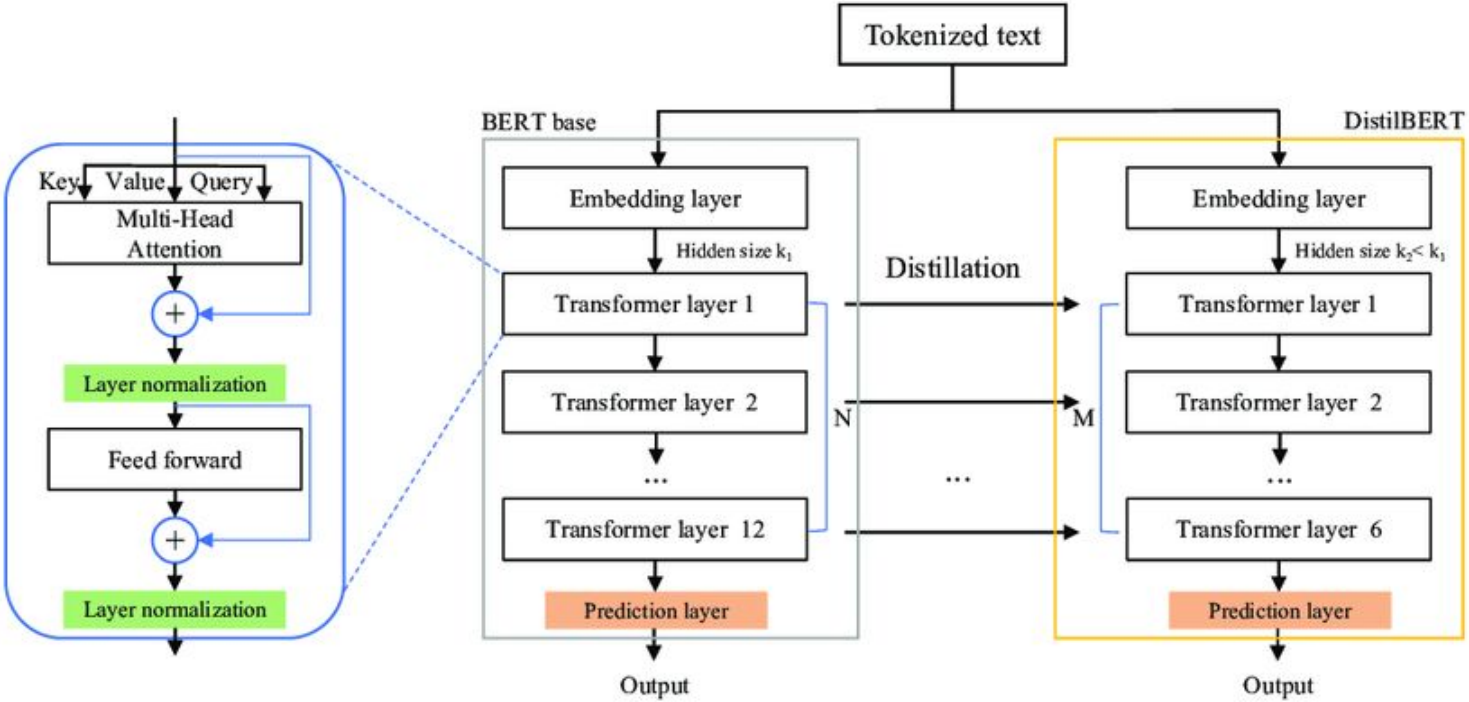




MS3

- Using DistilBERT: a Pre-Implemented Model
- Pre-Processing the Data for the Chosen Model
- Training the Model on the Dataset
- Evaluating the Model and Post-Processing

DistilBERT Model Architecture



Preparing the Data

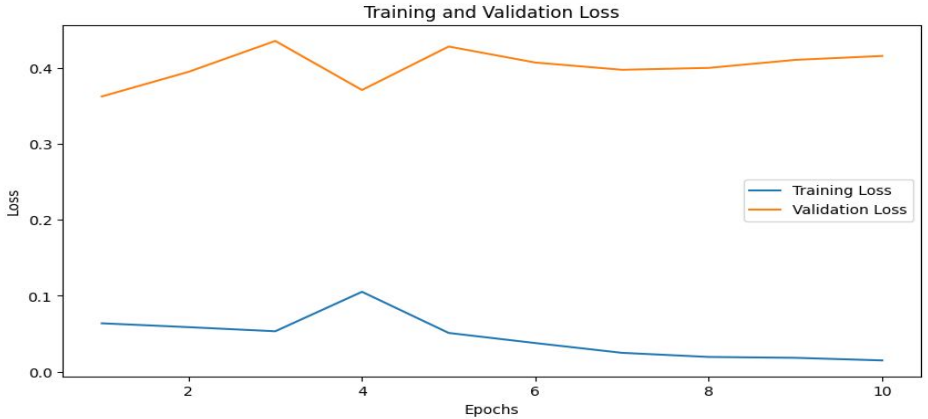
- The text data was tokenized using DistilBERT's tokenizer.
- Attention masks were created to inform the model which tokens should be attended to and which should be ignored.
- The tokenized text was padded using the DataCollatorWithPadding class to ensure batches have the same length.
- The Triage class organizes the preprocessed data into a specific format required for training with DistilBERT, encapsulating the tokenized inputs, attention masks, and target labels within a dictionary.

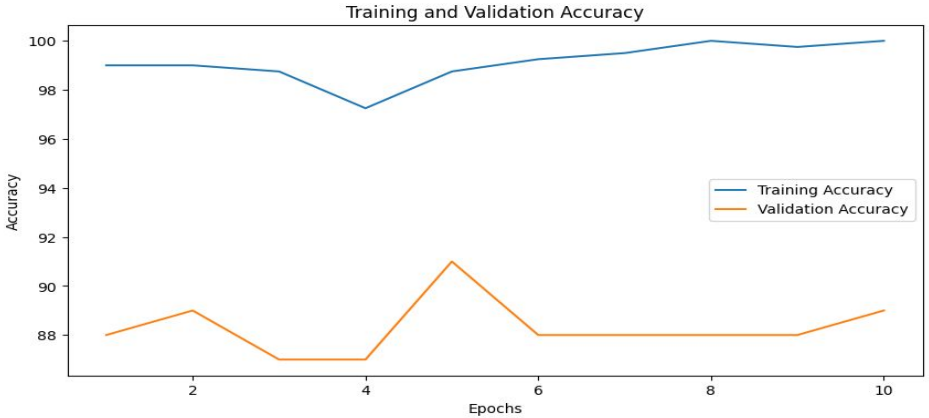
Training the Model

- The Adam optimizer and a learning rate scheduler were used to manage the learning rate during training.
- For each batch, the input IDs and attention masks were prepared and moved to the device.
- The model performed a forward pass to obtain the outputs, and the loss was calculated using a predefined loss function.
- Intermediate training loss and accuracy were printed every 5000 steps to monitor progress.
- Backpropagation was performed, and the model parameters were updated using the optimizer.

Testing and Evaluation

- The model's performance was tracked over epochs, recording training and validation loss and accuracy.
- Plots of training and validation loss showed a decreasing training loss but a stable or increasing validation loss
- Training and validation accuracy plots revealed a significant gap, with high training accuracy but lower and fluctuating validation accuracy.





Analysis and Conclusion

Analysis of LSTM model

- High training accuracy, poor validation accuracy.
- Minimal overfitting observed.
- Dataset size: only 500 rows.

Future Directions for LSTM

- Investigate alternative model architectures.
- Explore simpler models.
- Consider dataset augmentation techniques.
- Incorporate additional data samples.
- Aim to enhance model generalization and performance.

DistilBERT Model Analysis

- Complexity reduced compared to BERT, but still challenging.
- Fine-tuning hyperparameters specific to the task.
- Further optimization required for effective learning.
- Potentially limited by dataset size.

Future Directions for DistilBERT

- Try Regularization and Normalization techniques.
- Modify model architecture.
- Enhance dataset with additional samples.
- Explore Domain specific pre-training.

Comparative Analysis and Insights

- LSTM: Interpretability, long-term dependencies, overfitting.
- DistilBERT: Transformer efficiency, overfitting, dataset size issues.
- Aligning model complexity with dataset characteristics is crucial.
- Iterative refinement of architectures and augmentation strategies.
- LSTM and DistilBERT face similar challenges.
- Importance of dataset size and quality.

Conclusion

- LSTM and DistilBERT show promise with High Accuracies but have limitations.
- Overfitting and generalization issues need addressing by model refinement or data augmentation.

Thank You....

Q&A