Task 3: Sentiment analysis using a single hidden layer RNN

Hyper parameters: Learning rate - 0.001 Epochs - 10 No. of Layers - 1 No. of nodes - 25

Experiment and Observations:

- An instance of the raw dataset is given below. It has a lot of emoji icons , punctuations and words beginning with hashtag. The first step Involved cleaning the dataset. the cleaned dataset is given below.
- A single layer RNN is built and trained using the given data. The train and test accuracy was similar throughout the training epochs. But the loss was high. The F1 score was very low
- This raised some suspicions over the dataset. When viewed, it is revealed that the dataset has about 90% negative sentiment examples.
- The model fit the training data and was always giving out Negative as the output no matter what the input is. This is the reason for high accuracy with high loss.
- The dataset is later tweaked such that both Negative and Positive samples are equal in size and then trained.
- The model trained on this tweaked data got less accuracy but was able to generalize better. This is inferred from a much lower loss value and high F1 compared to the previous attempt.
- 200-dimensional GloVe embedding is used as the word embedding. The words are padded and indexed before embedding.
- The evaluations are done and the plots are listed below

	id	sentiment	tweet
0	10120	Negative	i am composed. #i_am #positive #affirmation
1	20323	Positive	@user i repeat: very concept of #blackvote, #h...
2	31334	Negative	spring water! #peace #love #organic #vegan
3	849	Negative	@user @user @user rightly so! gop hates trump ...
4	5050	Negative	aww yeah it's all good bing bong bing bong

Raw Data

	id	sentiment	tweet
0	10120	Negative	i am composed i am positive affirmation
1	20323	Positive	i repeat very concept of to only talk about
2	31334	Negative	spring water peace love organic
3	849	Negative	rightly so trump more than
4	5050	Negative	yeah it s all good bing bong bing bong

Processed Data

```
label
0    1311
1      89
Name: count, dtype: int64
```

Train Data

```
label
0    396
1    396
Name: count, dtype: int64
```

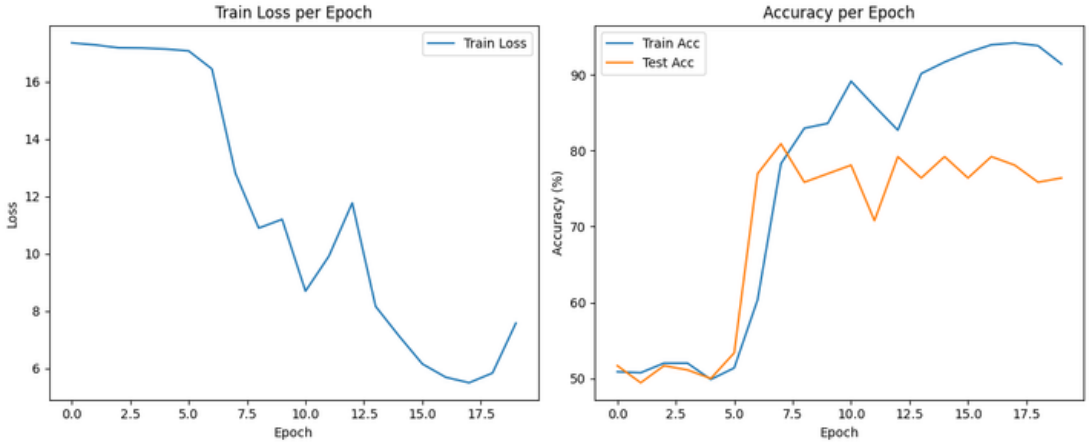
Tweaked Train Data

```
label
0    5204
1     396
Name: count, dtype: int64
```

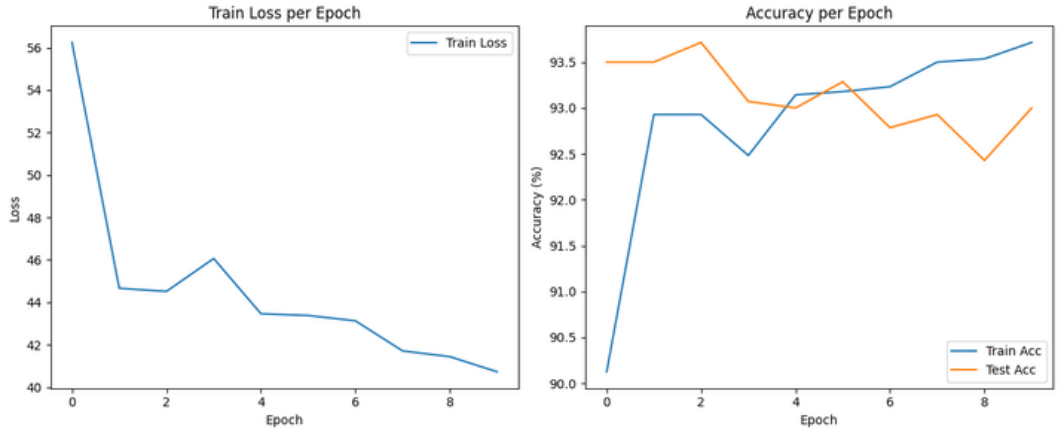
Test Data

```
label
1    89
0    89
Name: count, dtype: int64
```

Tweaked Test Data



Tweaked data Training plots



Untweaked data Training plots

Observations for Untweaked data

- **Epoch** - 10
- **Loss** - 40.7585
- **Train Accuracy** - 93.71%
- **Test Accuracy** - 93%
- **F1 Score** - 0.23

Observations for Tweaked data

- **Epoch** - 20
- **Loss** - 7.5858
- **Train Accuracy** - 91.41%
- **Test Accuracy** - 76.40%
- **F1 Score** - 0.74

Inferences made:

- The experiments reflects accuracy paradox—where a model achieves high accuracy simply by predicting the dominant class, but fails to capture the true performance, especially in imbalanced datasets.
- After balancing the dataset with an equal number of positive and negative samples:
- The accuracy dropped slightly (as the model could no longer rely on majority class bias),
- But the loss reduced significantly, and The F1 score improved drastically to 0.74, indicating better generalization and balanced predictive performance
- This demonstrates that: Balanced datasets are crucial for classification tasks, especially when using metrics like F1 score which account for both precision and recall.
- Relying solely on accuracy in imbalanced scenarios can be misleading.
- Furthermore, the use of 200-dimensional GloVe embeddings provided meaningful word representations, and padding/indexing allowed consistent input for the RNN model.