**Report**

**Group-31**

Submitted by - Nishil Agarwal(2022334)

Keshav Chhabra(2022247)

Himang Chandra Garg(2022214)

**Aspect Term Extraction**

**Explanation of preprocessing steps:**

Pre-processed the dataset by breaking down each sentence in dataset into tokens and applying BIO encoding to it to get a list of labels.

This was achieved as follows :

* Wrote code to break sentence at each space. Stored the token, starting index of token and the ending index of token in a tuple. (to use to and from indices provided in dataset later)
* For each stored tuple, extracted tokens from it to get tokenised form of a sentence.
* First labelled all tokens as ‘O’. Then, for each aspect word’s to and from indices provided in dataset, marked tokens with B or I if their starting index lied in the to and from range. Handled extra token start/end index alteration due to punctuation too.

If the previous label was not B, we would label the token with B else I.

* Created a vocab with <PAD>,<UNK> tokens along with dataset tokens obtained by breaking sentence as spaces.
* Created CustomDataset class to serve as dataloader. Padded sentences with <PAD> token so that all sentences are of same length. This is required for RNN and GRU models to work properly.

Labelled all these <PAD> tokens with -1 so that these can be ignored later during F1 score/Val loss calculation.

**Model architectures and hyperparameters used:**

RNN models:

Layer 1 : Embedding layer. Converted word encoding to embedding as per embedding matrix of respective model.

Layer 2 : RNN layer. Input size = embedding size. Output size = hidden size. Kept this 128 for Glove due to smaller size embeddings. And 256 for Fasttext as it has much larger sized embeddings.

Layer 3 : Fully Connected Layer. Input Size = hidden size. Output size = 3. As we have to classify all points into 3 categories B,I,O.

GRU models:

Layer 1 : Embedding layer. Converted word encoding to embedding as per embedding matrix of respective model.

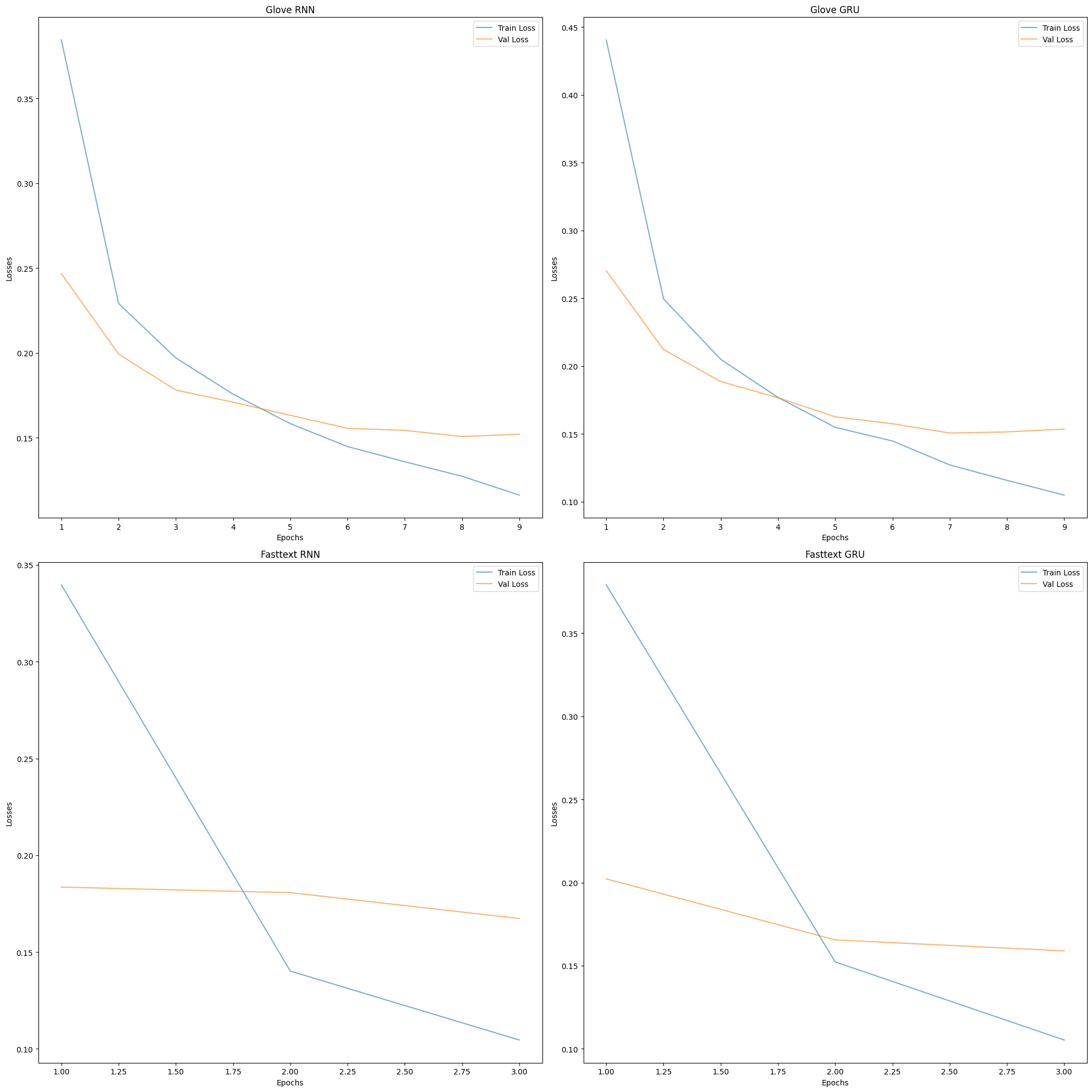
Layer 2 : GRU layer. Input size = embedding size. Output size = hidden size. Kept this 128 for Glove due to smaller size embeddings. And 256 for Fasttext as it has much larger sized embeddings.

Layer 3 : Fully Connected Layer. Input Size = hidden size. Output size = 3. As we have to classify all points into 3 categories B,I,O.

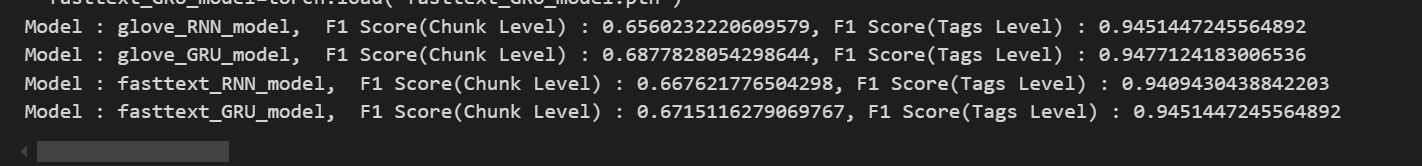
Kept Learning rate of 0.001 in all cases.

Trained for 9 epochs in case of glove models and 3 for fasttext. This is because the model seemed to overfit beyond these points. The Val loss continues to increase/fluctuate while train loss kept on going lower.

**Training and validation loss plots:**



**Performance comparison of all models:**



We can see that GRU models in both cases perform somewhat better. Especially in the case of Chunk scores. RNN models struggle with keeping long term dependencies in sentences due to lack of reset/forget gate mechanisms. They suffer from vanishing and exploding gradient problems. While in tag-level scores short term memory is enough, for chunk-level scores where multiple words are constituting aspect term, longer dependencies are useful. Hence, GRU one’s performed better.

Also, we find Fasttext scores and Glove scores quite similar. This is because unlike Glove, Fasttext uses sub word representations. This would in general perform better. But in our case, we created our own vocabulary from the corpus due to which Fasttext sub word representation did not really get a chance to shine as much due to lack of significant amount of OOV words (out of vocabulary).

**Best-performing model and its evaluation:**

Best performance observed in Glove GRU. With F1 score at chunk level = 0.6878

And F1 score at tag level of 0.9477.