NLP HW#3

Student Name:
Mehdi Moosavi

SID:
810102264

May , 2024

Dept. of Computer Engineering

University of Tehran

Contents

1 Preprocessing

The data at hand consisted of tokenized sentences with corresponding labels, divided into four parts: tokenized sentences, the index of the verb in the sentence, the label of each token in the Semantic Role Labeling (SRL) task, and the index of the word. For preprocessing, we implemented specified methods and applied them to both the validation and training datasets. We used a cutoff frequency of 2, a maximum vocabulary size of 20,000, and removed about 30% of less frequent items. Sentences with a length less than 50 were padded to meet this limit.After loading the training set, we used the obtained 'vocab' class to convert validation sentences into tensors, enabling the training and evaluation of the model on both datasets.

2 LSTM-based Model for Semantic Role Labeling

For predicting semantic role labels, we implemented a single-layer LSTM model using PyTorch. The model architecture followed the specified design:

Neural Network Architecture:

1. We passed each word's embedding vector through an LSTM layer and obtained the corresponding hidden state of the LSTM.
2. The output of the verb was also obtained.
3. We associated the output of the verb with the hidden state of each token.
4. Finally, we passed the outputs from the previous step through a linear layer to generate the final output.

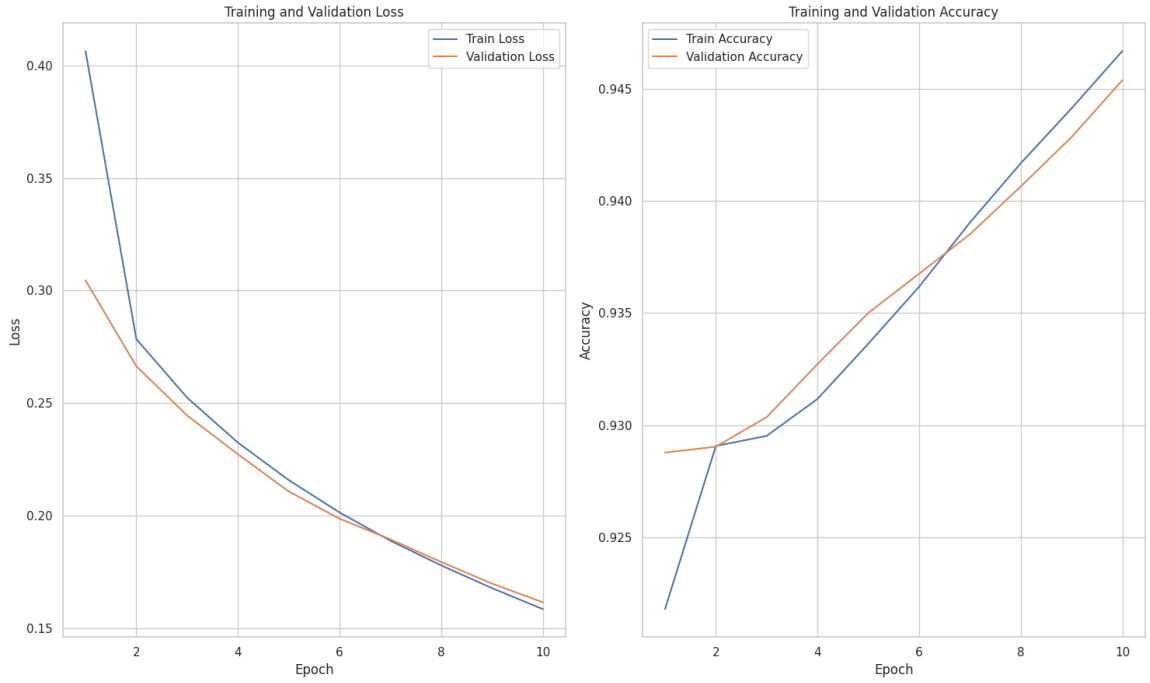

Figure 1: Training Loss and Accuracy for the LSTM-based model

As observed in Figure 1, training the model for 10 epochs resulted in favorable results. Moreover, considering the high accuracy achieved on the validation set, it's evident that the model performed well on this dataset.

The F1 score using micro averaging achieved a strong value of 0.9445, but when we switched to macro averaging, it dropped significantly to 0.5282. This decline indicates a notable imbalance between classes within the dataset, mainly because there are a lot of 'O' labels due to padding. Additionally, the frequent occurrence of 'O' as the most repeated semantic role in the initial sentences worsens this imbalance.

3 GRU-based Model for Semantic Role Labeling

3.1 GRU model and training

Continuing our exploration of models for semantic role labeling, we now turn our attention to a single-layer GRU architecture implemented using PyTorch. This model, like the LSTM-based approach discussed earlier, aims to predict semantic role labels through a series of neural network layers and the structure of it is exactly like the LSTM we implemented earlier.

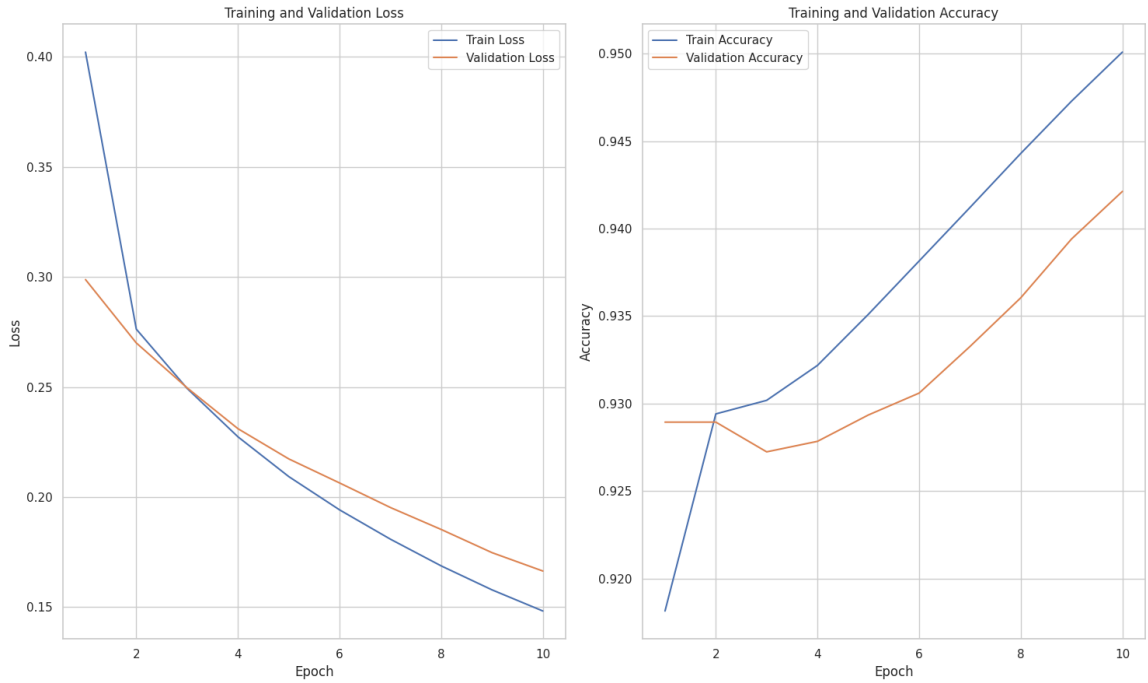

Figure 2: Training Loss and Accuracy for the GRU-based Model

After training the GRU-based model for 10 epochs, it showed promising results comparable to those of the LSTM model. Furthermore, the validation set's high accuracy reinforces the effectiveness of this model architecture.

The results are also close to what we had in previous section and F1 score on the test set with micro averaging is 0.9504 and using macro averaging it increases to 0.5504

3.2 LSTM vs GRU

1. **What is the advantage of LSTM over GRU?** LSTMs are better at capturing long-term dependencies in sequential data due to their additional memory cell and gated structure, allowing them to retain information over longer sequences more effectively than GRUs.

2. **Difference between LSTM and GRU:**

   LSTMs have separate input, forget, and output gates along with a memory cell, while GRUs have a combined update and reset gate without an explicit memory cell, resulting in a simpler architecture. Additionally, GRUs are generally faster to train due to their simpler architecture, while LSTMs tend to perform better on tasks involving long-term dependencies and complex sequential patterns.

3. **Why do we need to concatenate the output of the verb with the output of all tokens?**

   Concatenating the output of the verb with the output of all tokens allows us to access information related to the verb, providing better insight into identifying roles such as agent, patient, and others. Verbs typically have semantic or grammatical relationships with these roles, making it essential to incorporate their information for more accurate role labeling.

4. **Vanishing gradients solution** Utilizing checkpointing by splitting long sequences into shorter subsequences during training can help the vanishing gradient problem. Additionally, applying gradient clipping to limit the magnitude of gradients during backpropagation is another effective technique. However, it's worth noting that while gradient clipping may prevent vanishing gradients, it may lead to worse performance for learning long dependencies or relations.

4 SRL using encoder-decoder

4.1 Model and training

In the next problem we aim to tackle the SRL task via an seq2seq approach using a encoder-decoder model with attention.As mentioned in the problem the first step was to create the new dataset suitable for question answering task. the maximum length found for the questions was 53 and for the answers this was 16.

After loading the dataset considering the given instructions, first part of the model which was implemented was encoder which was consisted of an embedding layer, a LSTM layer and for the last layer a linear one.For the decoder too we used relatively similar architecture but in the decoder we have attention layer output too which will be concatenated to find more nuances in the data.

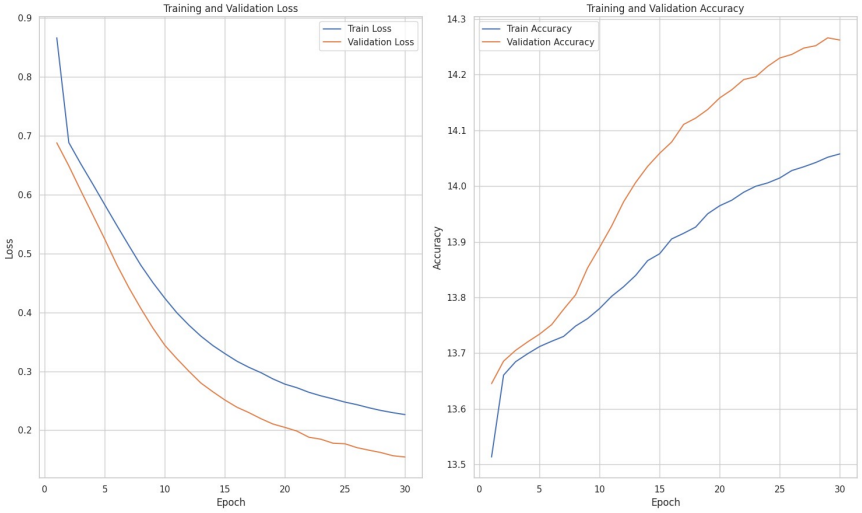

Figure 3: Plot of Loss and Accuracy During Training for the Encoder Decoder Model

After training the model we reached the F1 score of 0.9508 for the micro averaging and 0.5763 for macro averaging which is slightly better than previous results.

4.2 Seq2Seq using encoder-decoder

1. **What are the limitations of transforming a SRL task to the question answering ?**

   Question answering is usually a harder task to accomplish, and changing the task from SRL to SRL using QA models may lower the metrics. Additionally, the dataset needed for QA is more complex. In the task at hand, we had structured questions, which made the task easier.

2. **Why do we use 'START' and 'END' tokens in answers?**

   By using these tokens, we initiate the model's answer sequence with the 'START' token. When it's necessary to conclude the sequence, the model returns the 'END' token, indicating that we've reached the end of the sequence.

5 Analysis

In the Table 1 We can observe the model's performance across four different inputs. The question column is constrained to 25 tokens to fit within a table format. As previously mentioned, in this task, information about verbs is crucial for identifying roles. However, when transitioning to a question-answering format,it seems that the model struggles to grasp the significance of role tokens provided within questions. Overall, we observed that the question-answering method yielded better metrics.

Table 1: Sample Outputs

Question	Answer	Prediction
['coordinated', '<PAD>', 'The', 'progress', 'of', 'this', 'coordinated', 'offensive', 'was', 'already', 'very', 'entrenched', 'by', 'then', '.', 'B-ARG1', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>']	['<START>', 'offensive', '<END>', '<PAD>', '<PAD>']	['offensive']
['are', '<PAD>', 'What', 'you', 'are', 'interested', 'in', 'is', 'exactly', 'what', 'our', 'focuses', 'are', '.', 'B-ARG2', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>']	['<START>', 'interested', '<END>', '<PAD>', '<PAD>']	['interested']
['learned', '<PAD>', 'Well', ',', 'from', 'the', 'information', 'and', 'the', 'situation', 'you', 'have', 'learned', ',', 'how', 'would', 'you', 'two', 'interpret', 'some', 'messages', 'sent', 'out', 'by', 'their']	['<START>', 'you', '<END>', '<PAD>', '<PAD>']	['you']
['learned', '<PAD>', 'Well', ',', 'from', 'the', 'information', 'and', 'the', 'situation', 'you', 'have', 'learned', ',', 'how', 'would', 'you', 'two', 'interpret', 'some', 'messages', 'sent', 'out', 'by', 'their']	['<START>', 'the', '<END>', '<PAD>', '<PAD>']	['you']

However, these scores carry a different meaning compared to those in the second and third parts, where the output comprised 11 states. In contrast, the unique values in the output now span the entire vocabulary.

Additionally, as mentioned, there appears to be a challenge in extracting necessary information for Semantic Role Labeling (SRL) using the question-answering method, likely due to its increased complexity.