Natural Language Processing

NEURAL PART OF SPEECH TAGGING

Ashmit Chamoli

Introduction

This report contains training and testing details for neural PoS tagging architectures presented in this repository.

Two architectures are used in this report, based on ANN and LSTM respectively. The implementation details of both these architectures can be found in the README.

Dataset

For training, the Univeral Dependencies dataset has been used. Specifically, the file located in the `UD_English-Atis/` folder are used. For parsing this dataset,the conllu library has been used.

Training and Evaluation

ANN PoS Tagger

To tune the hyperparameters for this model, a grid search was performed on the following sets of hyperparameters:

- Embedding Sizes: 128, 256, 512
- Hidden Layers: [], [32], [64], [128], [128, 64], [64, 32]
- Activation: RelU, Sigmoid, Tanh
- Context Sizes: 0, 1, 2, 3, 4

All models were trained for 15 epochs with a batch size of 128 and a learning rate of 0.001. Evaluation metrics (accuracy, precision, recall and f1-score) were evaluated on the train set, dev set and the test set for each of the 270 models trained.

The following were the top 5 models based on dev set f1-score:

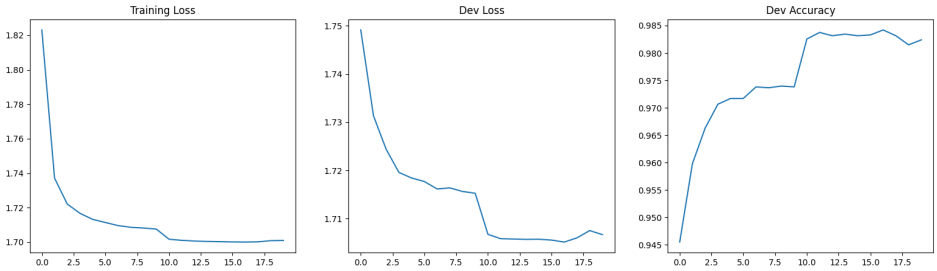
Rank	Hidden Layers	Embedding Size	Activation	Context Size	Dev F1
1	[64]	128	'sigmoid'	1	**0.98361**
2	[64]	256	'tanh'	1	0.98351
3	None	256	'relu'	1	0.98321
4	[64]	512	'sigmoid'	1	0.98319
5	[128]	256	'tanh'	2	0.98316

We can see that shallower models with sigmoid activation work better than deeper models (more hidden layers). But note that the difference between the top models is too minute to conclude anything confidently.

Following are the results for the best model on the test set:

- Accuracy: 0.98693
- Precision: 0.986886
- Recall: 0.98693
- F1-score: 0.986534

The following is the plot of training progress of the best ANN model:



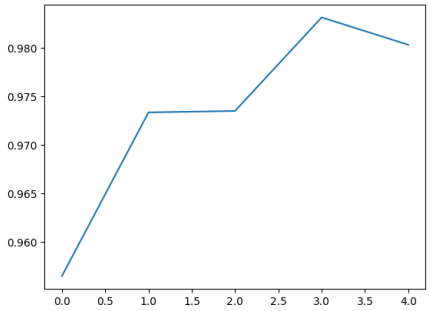
The plot shows various training metrics accross epochs. As expected all the metrics improve after each epoch.

This graph is expected. We will now look at the confusion matrix for the best model.

The confusion matrix for the best ANN model.

As is evident by the confusion matrix, the model is performing really well for most classes. However the model is slightly confused between adverbs and adjectives and predicting 22% of the adverbs as adjectives. This is an interesting observation because adverbs and adjectives are often difficult to tell apart even for humans. Another interesting observation is that the model almost never confuses adjectives for adverbs, that is, it almost always classifies adjectives correctly.

Next, we will plot accuracy vs context size for the best model.



The plot shows accuracy vs context size for the best ANN model.

We can see that the accuracy reaches a highest point after which it starts decreasing. With this, we can conclude that when the context is higher than this threshold, it just adds noise for the model which is a hindrance. On the other hand, if the context is less than this threshold, the amount of information is lower and hence the model does not perform well.

RNN PoS Tagger

To tune the hyperparameters for this model, a grid search was performed on the following sets of hyperparameters:

- Embedding Size: 128, 256
- Hidden Embedding Size: 64, 128
- Hidden Layers: [64], [32, 16], [64, 32]
- Number of Stacks: 1, 2, 3

For all the models, learning rate, epochs and activation was set to 0.001, 13 and RelU respectively.

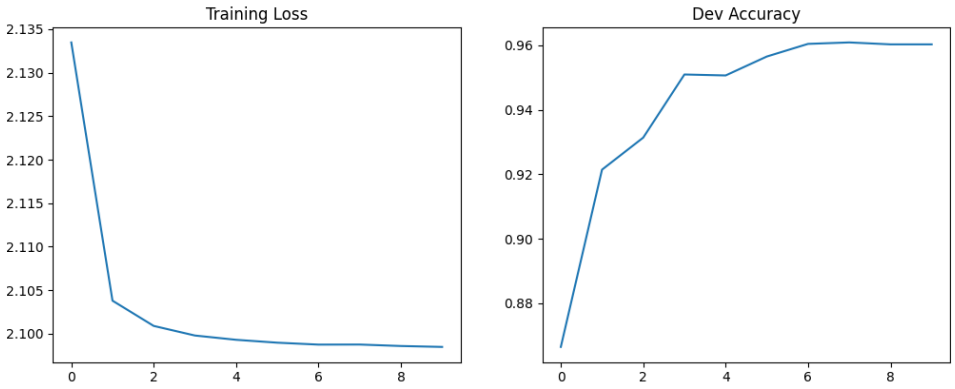
The top 5 models based on dev set f1-score are:

Rank	Hidden Layers	Embedding Size	Hidden Size	Num Stacks	Dev F1
1	[64, 32]	256	64	2	0.971081
2	[64, 32]	256	128	2	0.971079
3	[64]	128	128	2	0.971017
4	[32, 16]	256	128	2	0.970926
5	[32, 16]	256	128	3	0.97083

The following are the results of the best model on the test set:

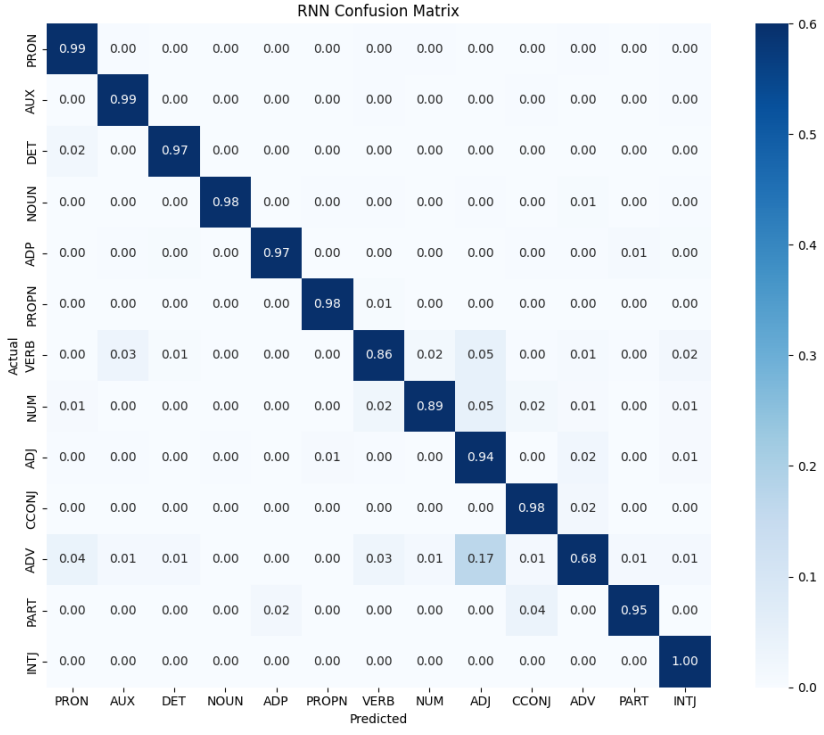
- Accuracy: 0.972796
- Precision: 0.974626
- Recall: 0.972796
- F1-score: 0.973238

The following is the plot of training progress of the best RNN model:



The plot shows various training metrics accross epochs. As expected all the metrics improve after each epoch.

This graph is expected. We will now look at the confusion matrix for the best model.

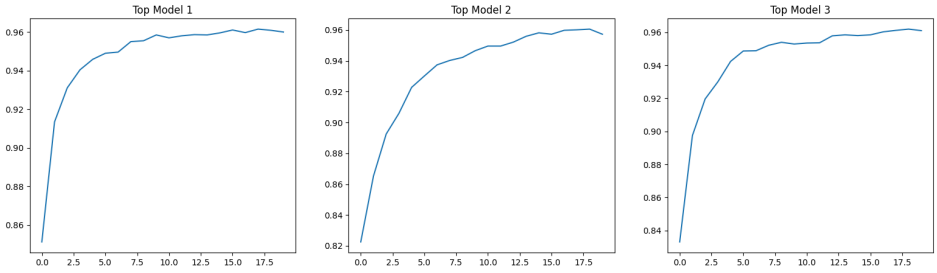


The confusion matrix for the best RNN model.

Again, the model is performing really well for most classes. We again observe that the model is confusing adverbs with adjectives but RNN slightly beats ANN (ANN confused this class 22% of the times while RNN did it only 17% of the times), even though the overall accuracy is a bit lower for RNNs. Similarly, the model is not confusing adjectives for adverbs in this case as well.

During experiments, it was also observed that LSTMs always performed better with bidirectionality.

Next, we will plot dev accuracy vs epoch for the top 3 models



The plot shows dev accuracy vs epochs for the top 3 models.

We can see that model 2 improves its accuracy a bit slower than the other 2 models. Model 1 achieves maximum accuracy the fastest and then remains stable.

Conclusion

Overall, the ANN Tagger performs slightly better than RNN Tagger. The reason could be that RNNs will require more data to be trained properly. The RNN Tagger would begin to overfit the data before the accuracy on the dev set could improve to become comparable to ANNs.

In any case, both the architectures have proven to be fairly good at the task of PoS tagging.