ASSIGNMENT 1 | NLP

Shamik Basu - 0001035358 | Pratiksha Pratiksha - 0001034021 | Pritikumari Gupta - 0001026995

MSc Artificial Intelligence, Unibo

ABSTRACT

The purpose of this assignment is to perform Part-of-speech(POS) tagging. POS tagging is an important NLP process which refers to categorizing words in a text (corpus) in correspondence with a particular part of speech, depending on the definition of the word and its context. In our approach we have used five different types of networks and parameters - Bidirectional LSTM, GRU, Baseline Model with additional LSTM Layer, GRU with additional LSTM layer, GRU with Bi-Directional LSTM layer. The relative performances has been analyzed by considering the overall qualities of datasets and to possibly introduce further improvements. Evaluated on validation sets the best two models "Bid-directional LSTM" and "GRU" are selected.

1 SYSTEM DESCRIPTION

The dataset provided was splitted in train, validation and test set accordingly to the indications given. To provide those sentences as input to Recurrent Neural Network (RNN) architecture, a tokenization for each train and validation set has to be provided for appropriate representation of the terms. In order to do so, word embedding has been used to represent terms in the dataset. A total of forty-five POS tags were found. Furthermore, Co-occurrence matrix was built to count the number of times each word appeared within the same context window. The Glove embedding model was implemented with dimension '300' and Out-Of-Vocabulary (OOV) terms were calculated. Compared to total words found in Glove the percentage of OOV words was equal to 5.22 % in training set and validation set. To build an embedded matrix we used sliding window approach. If an OOV words was found, a window size of $size \geq 1$ was considered and a weighted average of both the neighbour words vectors was taken from GloVe embedding (left and right) and was assigned while building the dense embedding. Where there was no left or right neighbour, a random vector was assigned to the OOV term. Padding is a special form of masking where the masked steps are at the start or the end of a sequence and it was executed to fix the sentence length. The dataset has largest sentence containing 250 tokens whereas MAX_SEQ_LENGTH is 100.

- Sentences longer than MAX_SEQ_LENGTH are truncated
- Sentences shorter than MAX_SEQ_LENGTH are padded with zeroes

All these steps were repeated for training and validation set. Before using RNN, one-hot encoding has been used for the output padded y-vector and the X-padded input dataset was split as per the length of sentences in each dataset found in the beginning of the code.

2 EXPERIMENTAL SETUP AND RESULT

As required by the assignment, a baseline model with Bidirectional LSTM and a Dense/Fully connected layer on top has been defined. LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Four different models have been implemented other than baseline model and each one has a dense/fully connected layer on top as the classification head. Here a brief description of rest of the models is mentioned-

1. **GRU Model** : The second model is Gated Recurrent Unit (GRU). GRU has fewer parameters than LSTM and easy to train.
2. **Baseline+LSTM** : In the third model additional layer of LSTM is used on top of baseline model.
3. **GRU+LSTM** : In fourth model approach, a additional LSTM layer is implemented on top of GRU layer
4. **GRU+Bi-Directional LSTM** : Similarly, the fifth model takes the GRU model and it adds on top of the Bidirectional LSTM layer.

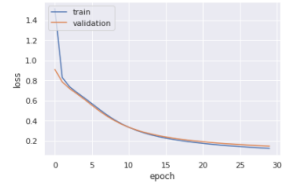
The first layer of all the models were a non-trainable Embedding layer which transforms the words in their corresponding embedding by taking the already predefined embedding matrix as input. The hyperparameters used are categorical_crossentropy as loss function to maximize the accuracy, Adam optimizer with default learning rate and a preferable metrics to obtain the accuracy and the batch size of 128. For each model the training was executed up-to 30 epochs included early stopping callback with 5 epochs of patience and a minimum accepted delta of 0.0001.
Finally the results of validation set were used to choose two best models based on f1-macro. The two best models were "bi-directional LSTM"(baseline) and "GRU".

Model	Train		Val		Test	
	Acc	F1	Acc	F1	Acc	F1
Baseline	.96	-	.96	.61	-	.61
GRU	.96	-	.96	.61	-	.60
Baseline+LSTM	.95	-	.95	.47	-	-
GRU+LSTM	.95	-	.95	.47	-	-
GRU+BiLSTM	.96	-	.96	.59	-	-

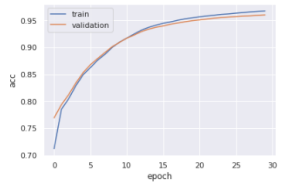
Then these models were evaluated on the test set where some experiments were carried out to understand the dataset and models better. Overall evaluation result for all the models are shown in above tabel. The best model (baseline model) result an F1 score of 0.61 on the test set. Here the F1 score on test set is shown with removal of zero support tags and punctuation class.

3 DISCUSSION

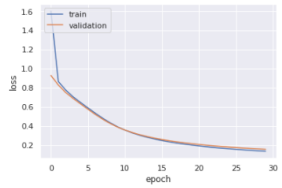
The difference between the model accuracy on the validation set was not too high. The two best models were used to carry out different experiments on the test set. On inspecting, we observed that the result on the best models quite convincing. The models were able to predict the same POS tagging as the test set. The baseline has the best performance on the validation set and gives quite impressive graph for loss and accuracy for train VS validation set. No overfitting was observed in any of the model and impressive curves were obtained. Below are the loss and accuracy graph on train and validation set for the Baseline and GRU models.



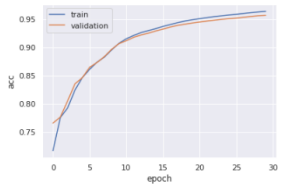
Baseline: Loss



Baseline: Accuracy



GRU: Loss



GRU: Accuracy

3.1 QUANTITATIVE RESULT DISCUSSION

It is observed that in the train set, the accuracy of "Baseline+LSTM" and "GRU+LSTM" is slightly lower than the rest 3 models. The other 3 models have the same accuracy of 0.96. The F1 score is not calculated on the train set while training as no gigantic mini-batch is considered. On the validation set "Baseline" and "GRU" has the highest F1 score of 0.61 and 0.61 respectively. Further, the two best models are evaluated on test sets. The F1 macro score of the baseline and GRU model is 0.61 and 0.60 respectively. It has been observed the result evaluated on the test set are consistent with the result of the validation set.

3.2 ERROR ANALYSIS

Different experiments were carried out on the two best models on the test set to achieve best result. This was beneficial in understanding the data and the performance of the models.

- It has been noticed that tags like NNPS and RBNS have a low score and some others like PDT, RP, RBR, LS, FW, UH, and SYM have zero F1 score. Most of these tags have no examples in the dataset. This is a major reason for thr poor prediction on the test sets.
- The classification report and the confusion matrix were computed on the test set for both the models and then the least frequent tags (zero support) were removed and the same metrices were calculated again. The tags with f1-score less than 0.3 were considered as the least frequent tags. It was observed that the f1 score of the GRU model was 0.60 and stayed the same even after removing the zero support tags whereas it was 0.54 for Baseline before removal and increased to 0.61 after removal of zero support tags. In every cases we excluded the punctuation classes.
- It is noticed that the dataset is skewed which is one of the main reasons of poor performance on test set. Maybe there could be improvement in the performance if more data or an evenly distributed dataset is provided to the networks.

4 CONCLUSION

We found Baseline model is the best model. To conclude, analysing the POS tagging errors, it is verified that the dataset was skewed which led to poor training of the RNN models. Choosing a different corpora or adding to the corpora sentences with the least frequent tags would improve the model generation. The limitation of the models are that they are unable to identify the missing tags in the test set and it is assumed that if some pre-trained models are used on bigger datasets, it will be able to successfully classify the missing tags in the test set.

References

[1] https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/dependency_treebank.zip

[2] ttps://towardsdatascience.com/pos-tagging-using-rnn-7f08a522f849