La Sapienza NLP Homework-2 2021

Benokan Kafkas
Matricola : 1850505
kafkas.1850505@studenti.uniroma1.it

1 Introduction

Topic of the second homework and this report is Aspect based sentiment analysis (ABSA). This task includes two tasks combined first of them is the aspect term identification which is first part and the aspect term polarity classification which is our second part which concludes aspect based sentiment analysis task.

2 Experiments

In the experiments there are several models which are shown in the graphs in the end of the report. But to summarize all models are using pretrained BERT(Devlin et al., 2018) as the base. And a second model for the aspect term identification is using CRF(Lafferty et al., 2001) (Conditional Random Fields) on top of pretrained BERT model, I've tried a similar approach to the paper of Bert for Portuguese Named entity recognition (Souza et al., 2019). For the second part all models that I'll show are using only pretrained BERT with fine tuning the BERT itself since it acquires better both macro f1 and validation scores. Both on part a and part b models are using a MLP with hidden layer and a final fully connected layer with dropout.

3 Preprocessing for Aspect Term Identification

As in the first homework the most challenging part of the project again is the preprocessing of the given data. Even the littlest mistakes are giving huge differences in the final result. Since aspect term identification is like a sequence labeling task first of all I've converted the the inputs to BIO tags. For example "I always use a backup hard disk to store important files." corresponds to " O O O O O B I O O O O" since "hard disk" is the our target word in the dataset. Since there are multiple types of target words which contain more than 1 word and multiple target words in one sentence using BIO was the first approach that came to my mind for ease of use. Also to locate the indices of the each target word I've seperated them with a whitespace to the beginning and also to the end because some of them were adjacent to the special characters and the bert tokenizer was producing faulty target words during tokenization.

Also during the preprocessing there were 3 datapoints in the laptops train file which were giving errors so I completely deleted them to feed the data to the model without any problems. It was obviously not the best approach but 3 data point in 2500 shouldn't change much in the results. As I mentioned above I've used bert-base-cased tokenizer to convert all the inputs to ids with special tokens.

4 Preprocessing for Aspect Term Polarity Classification

Approach in this task was a bit different since some of the sentences were including multiple target words the structure of the input that I fed to the network was also different.For the sentences that were including multiple target words I've created new data points (Same sentence but different target word). And there were also sentences which are not containing any target words and completely removed from the sentences that are fed to the network since they would not give any contribution to learning without any target words. The type of the tokenizer is still the same bert-base-cased for the inputs.

The challenging part of the second part was locating the target words right way. For that matter I've used a function called find targets which locates the target words in the sentences and basically return their index values respectively.

5 BERT Model

For the base of the both models for part one and part two I used BERT model to benefit from contextualized embeddings (pretrained bert base cased). I used two different approaches in the first part while using the model. First of them was to train the model with only using BERT and the second approach was to train the model by adding a CRF layer on top of bert embeddings and hidden + fully connected layers.

The model for the second part(Aspect Term Polarity Classification) to take advantage from the target word embeddings I've used a weighted average of the whole sentences with giving more weight to the contextualized target words' embeddings and fed it to the network that way. And all the models in part one are part two have one layer of hidden(nn.Linear(768, 768)) and one layer of fully connected nn.Linear(768, output-class-number)) in the end for the classification task.

CRF is basically a probabilistic discriminative model that is widely used in NLP tasks and computer vision task. It's used for predicting the sequences in my approach using contextual information to make correct predictions. Since I was using BIO tagging thus it was a sequence labeling problem using CRF was a common approach used by many researchers. I've used TorchCRF library to benefit from the loss function of CRF that gives a better prediction on output sequences.

6 Training and Evaluation

During the training I've tried different hyperparameters as I will show below in the figures section but the best performing approaches were with the 0.3 dropout for the first task and with the 0.7 dropout for the second task. Also the pytorch lightning module was used and it was pretty easy to train the model using the lightning module. Especially the tensorboard logs and the callbacks were making the training phase pretty easy compared to using raw pytorch.

In the evaluation part pytorch-lightning was coming handy again with inherited functions like on-validation-end which I've used to average all evaluation metrics like f1 scores of each class, average losses, average accuracies and average macro-f1 scores. To calculate the f1 scores first I've implemented a similar structure as in the evaluation.py folder that's written by our professor and TAs which you can find it as TENSOR-to-BIO function and also commented out in the SequenceTagging.py folder.

From the pytorch-lightning callbacks I've used Early stopping to avoid overfitting and ModelCheckpoint for saving the best validation accuracy in the models that I've trained.

For the optimizer the best performing was Adam for the BERT models. I've also used the AdamW which was suggested by the most but since it was so dependant to the learning rate it wasn't giving stable results and even sometimes it was not learning at all because of the weight decay.

Also during the training of the second part I've realised that fine tuning bert model instead of freezing it entirely with a low learning rate increased the results dramatically. For that part I've used a different learning rate for the both parts (in MLP part and in the BERT fine tuning part, 1e-4 abd 1e-5 respectively)

7 Results and Conclusion

As the results I found out that for the sequence labeling task freezing bert achieved better but in the second part fine tuning bert achieved better results. Also as I mentioned above the most challenging part of this homework was definetely the preprocessing part with the faulty prerocessing in the first couple of weeks I've achieved less than half of the accuracies that I've acquired now. A little index shift error can damage entire dataset and I've realized assertions are real life savers. The validation accuracies, losses and the related metrics will be shown in the figures section in the end.

In the end I guess this model is also improvable by using more advanced techniques for the model for example adding a BiLSTM in between Bert Embeddings and CRF layer could've worked better according to the papers but when I tried the BERT+BiLSTM+CRF (Wang et al., 2021) the model didn't fit to my gpu unfortunately. And also I've noticed that the model with the CRF should've performed better.Because for example in the restaurants data two models are giving results where model with CRF performs about %7 less than the model without CRF.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Fábio Souza, Rodrigo Frassetto Nogueira, and Roberto de Alencar Lotufo. 2019. Portuguese named entity recognition using BERT-CRF. *CoRR*, abs/1909.10649.

Yu Wang, Yining Sun, Zuchang Ma, Lisheng Gao, and Yang Xu. 2021. A hybrid model for named entity recognition on chinese electronic medical records. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(2).

8 Figures of Training

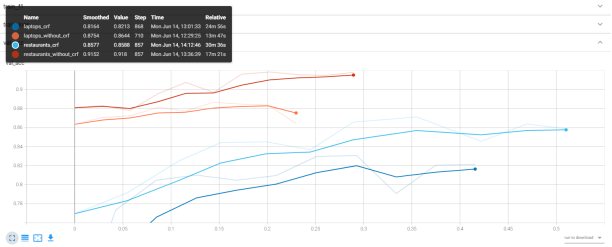

Figure 1: Validation Accuracy Part A

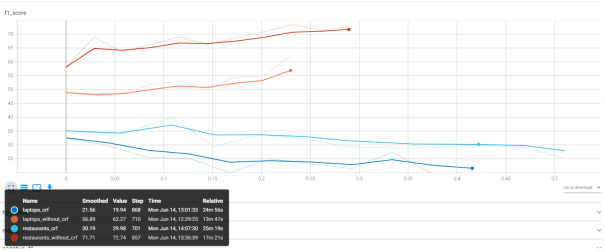

Figure 2: F1 scores for part A

[

Figure 3: Part B Macro F1

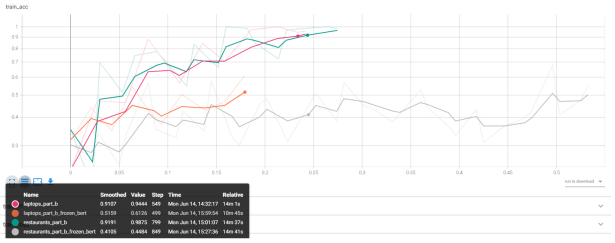

Figure 4: Train Accuracies part B

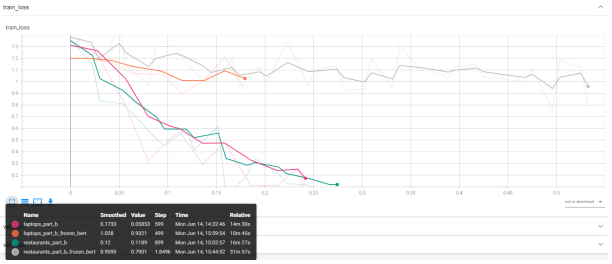

Figure 5: Training Loss Part B

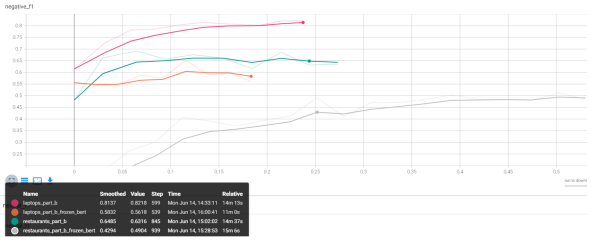

Figure 6: F1 of 'Negative' Label Part B

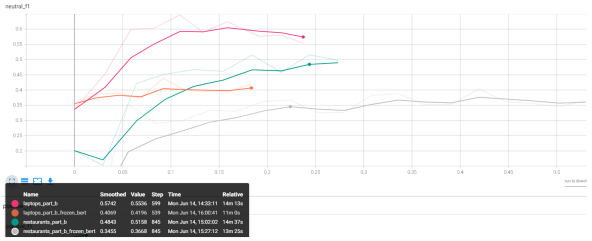

Figure 7: F1 of 'Neutral' Label Part B

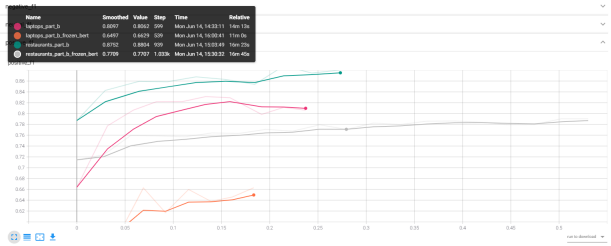

Figure 8: F1 of 'Positive' Label Part B

Laptops Part A	F1 Score	Validation Accuracy
Bert	0.56	0.86
Bert+CRF	0.25	0.82

Restaurant Part A	F1 Score	Validation Accuracy
Bert	0.717	0.918
Bert+CRF	0.37	0.86

Laptops Part B	Macro F1 Score	Validation Accuracy
Bert Fine Tune	0.67	0.69
Bert Frozen	0.48	0.508

Restaurants Part B	Macro F1 Score	Validation Accuracy
Bert Fine Tune	0.56	0.60
Bert Frozen	0.41	0.45

Figure 9: Results