Aspect Based Sentiment Analysis: Final Report

Group 2 (He Xinyi, Madeline Lim, Paul Tang, Tan Wei Zhen)

Abstract

Fine-grained Aspect Based Sentiment Analysis (ABSA) is a text analysis technique that identifies aspect terms and the respective sentiments attributed to them. In this study, we aim to explore potential enhancements to existing models through modifications to model layers as well as to overall model architectures. We find that model performance can be improved significantly by upstream embedding layer modifications, but only marginally by downstream neural layer modifications. Potential enhancements with cascaded models are also explored and discussed.

Introduction

Sentiment Analysis automates the mining of natural textual information to determine sentiment polarity of the sources, such as product reviews and customer feedback. However, such text data often contains more than a single general sentiment expression, particularly when said text has a balanced point of view. ABSA is thus desirable to extract the various aspects and expressions from a target topic.

Existing works have shown certain success with using BERT in the embedding layer of the models for Aspects Term Extraction (ATE) tasks and Aspect Term Sentiment Classification (ATSC) tasks. Further improvements were shown when sophisticated downstream neural layers were applied.

However, most experiments conducted adopt a single model to solve the two tasks end-to-end. It is in the interest of this report to explore how cascaded models, with one solving ATE and the other solving ATSC, would perform. We hypothesize that decoupling the two tasks would allow the models to focus on respective tasks and reduce performance dependency between operations. We also attempt further enhancement on end-to-end joint models proposed by recent related works, by exploring variations in the pretrained model for embedding layers and the downstream neural layers.

We find that change of models in the embedding layers plays a significant role in joint models' performance, whereas choices of the last layer of neural network appears to make marginal difference on performance. However, our results do not support the hypothesis that cascaded models would outperform joint models. Potential attributed factors are discussed.

Related work

A literature review conducted on past works indicates that prior to the arrival of pre-trained transformer models, most notably BERT in 2018, features were traditionally extracted using GloVe or Word2Vec for the embedding layer. However, these only provided a single context-independent representation for each token, limiting the performance of these models. In contrast, pre-trained models have shown to produce state-of-the-art results on NLP tasks due to the large amount of text these models have been trained on. Subsequent papers published after 2018 have largely relied on the use of such models in the embedding layer.

We refer to Li et al. (2019) as our key reference paper for this project. The paper aims to investigate the performance of BERT on the End-to-End Aspect-based Sentiment Analysis task by combining a BERT embedding layer with several variations of downstream neural layers, including linear layers, GRUs, and Self-Attention Networks. A key finding of the paper was that with even a simple linear classification layer, the paper's BERT-based architecture outperformed state-of-the-art works at the time. In addition, the introduction of more powerful layers (i.e., GRU, SAN) led to slightly better performance.

Approach

We explored both the joint and cascaded model approaches in this project.

Joint Models

For the joint model approach, we aimed to design a model of similar architecture as the paper mentioned above, as a control, experimenting with various downstream model types (i.e., linear, GRU) to determine the efficacies of each approach. The overall model architecture is illustrated in Figure 1 below.

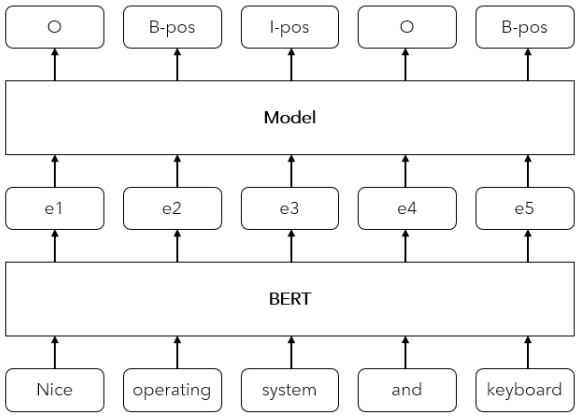

Figure 1: Joint Model Architecture

In addition, apart from using BERT as the embedding layer, we also experimented with the use of RoBERTa to determine the difference in model performance.

The loss function used is the cross-entropy loss over classes (O, B-pos, B-neg, B-neu, B-con, I-pos, I-neg, I-neu, I-con).

Cascaded models

For the cascaded model approach, our aim was to delegate the 2 subtasks to 2 separate models while also maintaining shared knowledge between the 2 model inputs, such that both models have the same general semantic and syntactic meanings from the pretrained BERT model. Hence, both models A and B take the same embedding inputs from a single BERT base. This also reduces the size of the model substantially as compared to using 2 BERT base models. The general structure of the cascaded model is shown in Figure 2 below.

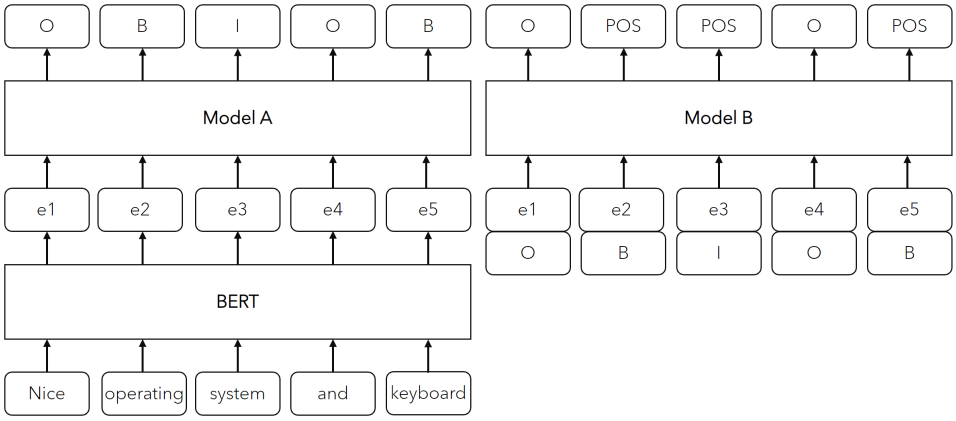

Figure 2: Initial Cascaded Model Architecture

The loss function used is the combined cross-entropy loss from the two subtasks.

Cross Entropy Loss A (CE_A) = Cross Entropy Loss over classes (B, I, O)

Cross Entropy Loss B (CE_B) = Cross Entropy Loss over classes (O, POS, NEG, NEU, CON)

Total Loss = $CE_A + CE_B$

Experiments

Dataset

SemEval is the International Workshop on Semantic Evaluation organised by the Association for Computational Linguistics. The SemEval 2014 and 2016 datasets for ABSA tasks consist of data on 2 specific domains, namely Laptops and Restaurants. In the datasets, each sentence is identified by a sentence ID, with its aspect terms and respective polarities annotated in XML tags by human experts. Possible polarity classes are 'positive', 'negative', 'neutral' and 'conflict'.

A summary of the datasets used are shown in the table below. The Laptops 2016 dataset was not used as no aspect term annotations were present in the data.

Dataset	Train	Test	Total
Restaurants (2014)	3,041	800	3,841
Laptops (2014)	3,045	800	3,845
Restaurants (2016)	2,000	676	2,676

For the joint models, pre-processing was performed on the datasets to first convert each review sentence to BERT- or RoBERTa-specific tokens. According to annotations given by the dataset, the tokens are then converted to respective lables according to their positions within a sentence. In terms of aspect-term classification, each word can be classified as outside-of-aspect (O), beginning-of-aspect (B), or inside-of-aspect (I). In terms of sentiment-classification, each word can be further classified as positive (pos), negative (neg), neutral (neu), or conflict (con). Hence, possible labels for a word in the sentence are O, B-pos, B-neg, B-neu, B-con, I-pos, I-neg, I-neu, or I-con.

For the cascaded models, further pre-processing was done on the joint model labels by splitting each label into its aspect term extraction label (B, I, O) and its aspect term sentiment classification label (O, pos, neg, neu, con).

Evaluation Metric

The models are evaluated based on their Micro-F1 score after excluding the O-labels. O-labels were excluded because they are not the main target for identification. The fact that O-labels naturally make up the main bulk of a sentence could skew target performance we are interested to investigate. We opted to use Micro-F1 score excluding O-labels over Macro-F1 score to place equal importance on each non-O token in the test set.

Experimental Details

Joint Models

Three joint models are implemented, namely BERT-Linear, BERT-GRU and RoBERTa-Linear.

For the BERT joint models, the smaller pretrained BERT model – bert-based-uncased – is adopted as the embedding layer. This pre-trained model comes with 12 layers, 768 hidden size, 12 attention heads and a total of 110 million parameters. Dropout of 0.5 is implemented for regularisation on the embedding layer, followed by a linear layer for the BERT-Linear model, or by a GRU layer with hidden-size of 512 and finally a linear layer for the BERT-GRU model. AdamW is used as the optimiser with a learning rate at 5e-5. The models are trained for 4 epochs on the training set, tuned on the validation. Predicted label classifications on the test set are evaluated against actual labels.

For the RoBERTa model, roberta-base is used, which contains 12 layers, 768 hidden size, 12 attention heads and 125 million parameters. The downstream model architecture is similar to the architecture described above for the BERT joint models.

Cascaded Models

Cascaded Model v1: Linear layers for A and B

Using linear layers for both models A and B, the model does not train well for both subtasks. While the train loss decreases, visualising the predicted BIO labels produced by model A shows that the model converges to predicting a single label, as shown in Figure 3 below. This was experimented with different parameters such as applying class weights to the cross-entropy loss, which only changed which label it converged to.

	0	1	2
0	0	0	5714
1	0	0	233
2	0	0	157

Figure 3: Cascaded Model v1 ATE Confusion Matrix. 0, 1, 2 represents O, B, I respectively

We hypothesised that due to models A and B being shallow linear layers, the Bert base model weights had to optimise over both loss functions while there was only a single linear layer trained specifically for ATE and ATSC each. This indicated that models A and B needed more complex layers to model each subtask well. We tested this hypothesis by making changes as seen in the next model.

Cascaded Model v2: Bert Encoder Layers for A and Linear layer for B

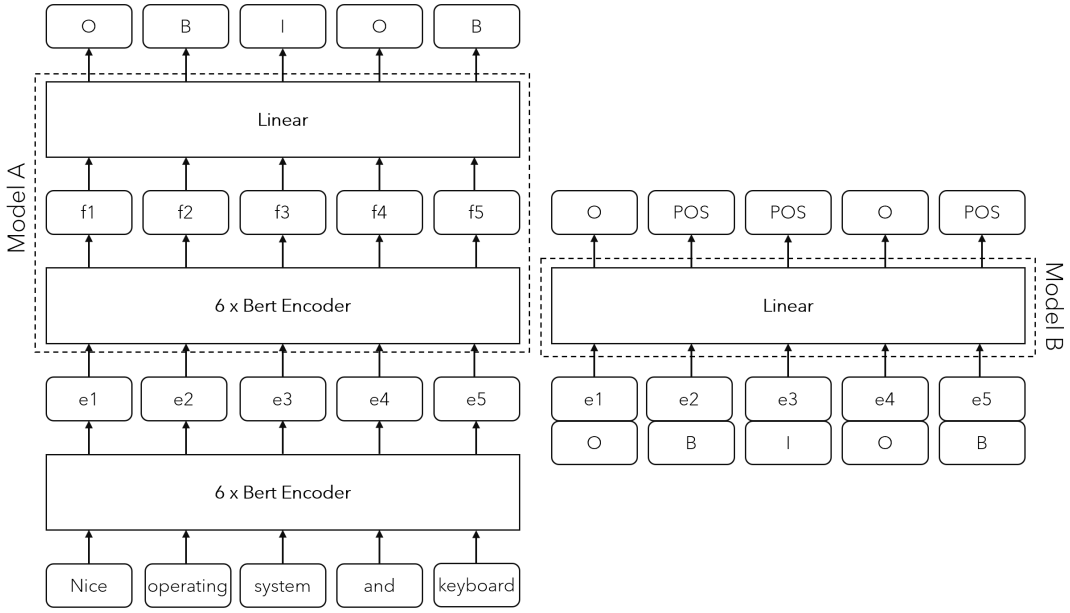

Figure 4: Cascaded Model architecture 2

To introduce deep layers into Model A while maintaining the total number of parameters, we used embeddings in the middle of the full Bert-base and used those as inputs to Model B. This allows the upper layers in the Bert model to train specifically for the ATE subtask.

	0	1	2
0	10844	161	179
1	89	514	11
2	129	22	447

	0	1	2	3
0	10928	254	2	0
1	218	778	0	0
2	69	86	5	0
3	18	38	0	0

Figure 5: Cascaded Model v2 ATE Confusion Matrix (Left) and ATSC Confusion Matrix (Right)

Still, to complete the hypothesis mentioned earlier, we should implement deep layers in both Model A and Model B such that both tasks are modelled by sufficiently complex networks. With improvements in results for the ATE subtask, it is expected that doing the same for Model B will result in improvements for the ATSC subtask.

Cascaded Model v3: Bert Encoder Layers for A and Transformer Decoder layers for B

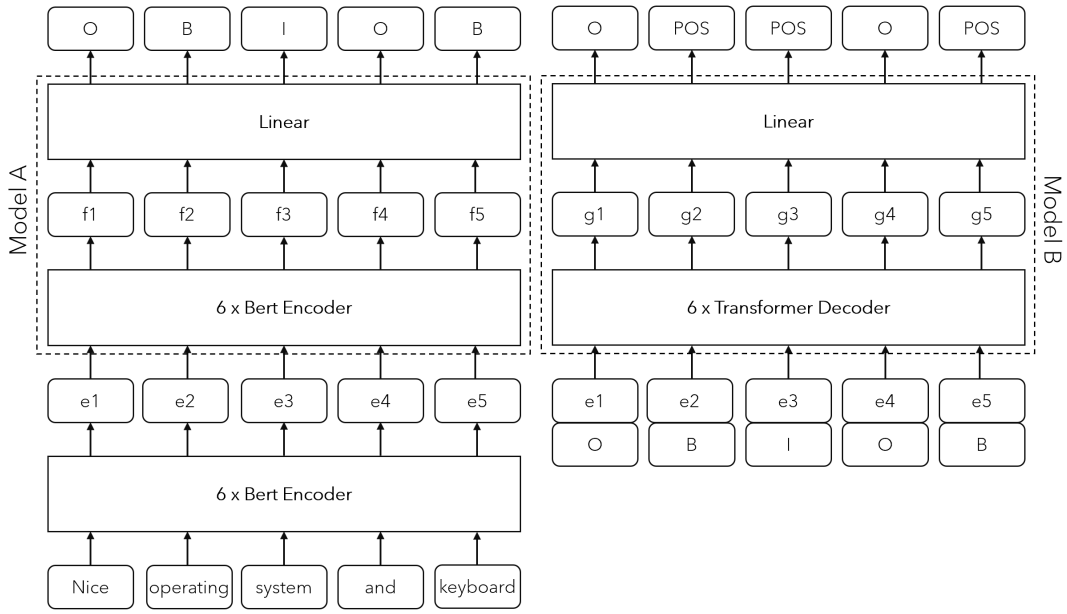

Figure 6: Cascaded Model architecture 3

To introduce deep layers to model B, we had to increase the total number of parameters in the model. To mirror a similar structure seen in model A, we use transformer decoder layers using the BIO embeddings as queries and the shared embeddings from the shared Bert Encoder layers as keys and values. The transformer decoder also does not used masked attention as we are able to see both future and past tokens in the sequence. This model refers to the GRACE model seen in Luo et al. (2020).

	0	1	2
0	10910	139	135
1	128	457	29
2	150	41	407

	0	1	2	3
0	10901	225	58	0
1	143	830	23	0
2	59	13	88	0
3	23	16	17	0

Figure 7: Cascaded Model v3 ATE Confusion Matrix (Left) and ATSC Confusion Matrix (Right)

As shown in the confusion matrices above, by implementing deep layers in both model A and B, the model was able to train well for both ATE and ATSC subtasks.

Results

Final micro-F1 scores of the various models, namely BERT-Linear, BERT-GRU, RoBERTa-Linear and the Cascaded model are shown in Figure 8.

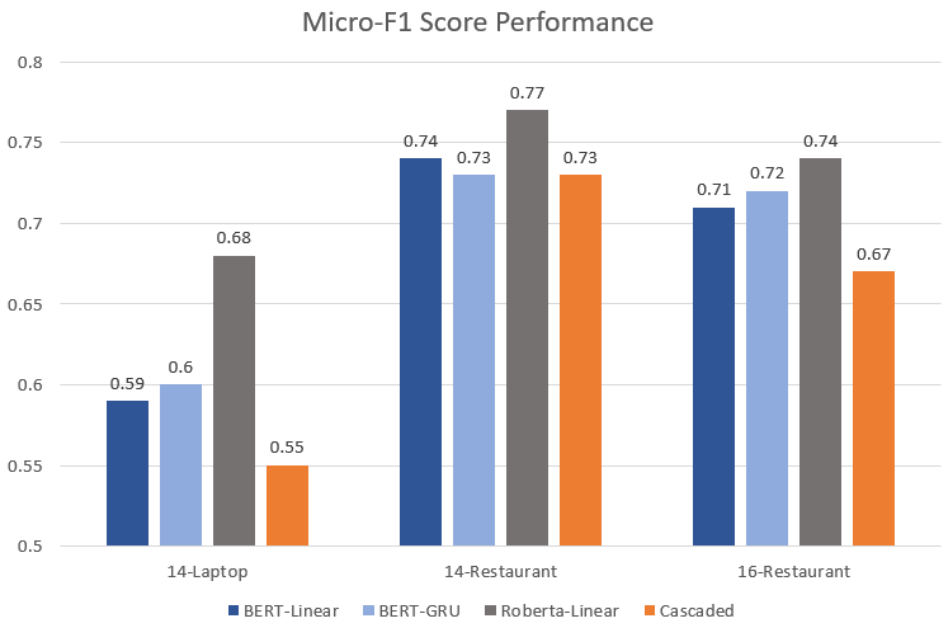

Figure 8: Micro-F1 Score across different models

Analysis

Joint Model Analysis

Pretrained Model selection

The RoBERTa-Linear model was able to show significant improvement in model performance over both BERT joint models. This result is largely expected, as RoBERTa has been shown to outperform BERT in numerous other NLP tasks, due to the improvements made in the pre-training stage of the model.

Final Layer Model selection

The last layer modification in the BERT joint models does not show significant effect on model performance, whereby BERT-GRU is not able to produce consistent improvement over BERT-Linear as one would expect. These results are largely aligned with the results obtained in the key reference paper, where more complex downstream models showed only very slight improvements in performance over the linear layer.

A possible factor is that the output from transformer BERT does not train well as input to an RNN layer GRU. However, BERT's bidirectionality is effective in capturing the context of original sentences and is therefore able to produce good results simply with a linear activation layer for final output.

Cascaded Model Analysis

Model A / B selection

As shown in the cascaded models experiments using deeper layers for both models A and B led to improved performance and training, while using shallow layers has shown to result in poor convergence in both subtasks.

Overall comparison in results between the joint models and the cascaded models shows that the joint models outperform the cascaded model. Despite better performance reported in the paper by Luo et

al. (2020), our implementation performed less well due to various factors. Our implementation lacked many enhancements mentioned in the paper, such as Post-training on Bert, Virtual Adversarial Training, and using a custom Gradient Harmonized loss function. Additionally, our implementation could have contained errors, and the hyperparameters chosen were not fine-tuned to a large extent.

Conclusion

From this study, we conclude that the embedding layer of a classification model plays a significant role in model performance. The choice of a more sophisticated pre-trained model such as RoBERTa would outperform a BERT-base model. We also show that a downstream GRU layer does not necessarily improve model performance over a linear layer. It remains to be discovered further if other modifications downstream could yield better results.

For the cascaded models, we have shown the difficulties in implementing such a model, where the models for both subtasks operating on a separated cross-entropy loss requires that the models specific to each task need to be complex enough for the model to train well. More complex training methods are also required to obtain good results, as Luo et al. (2020) has shown to use training of different parts of the model in multiple stages.

References

Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019). https://doi.org/10.18653/v1/d19-5505

Luo, H., Ji, L., Li, T., Duan, N., Jiang, D. (2020). GRACE: Gradient Harmonized and Cascaded Labeling for Aspect-based Sentiment Analysis. Findings of the Association for Computational Linguistics 2020. https://arxiv.org/pdf/2009.10557v2.pdf