Aspect-based Sentiment Analysis using Transformers

Group 507

Constantin Gabriel-Adrian
gabriel.constantin13@s.unibuc.ro

Sociu Daniel
daniel.sociu@s.unibuc.ro

Abstract

In this technical report, we will explore different variations of transformer based architectures for solving Aspect-based Sentiment Analysis (ABSA) tasks. Sentiment Analysis involves finding the overall sentiment of a text. By contrast, an ABSA task involves predicting the sentiment of a specific paragraph when given a context / term (i.e aspect-based). As such, we test different BERT-based architectures by fine-tuning them on the SemEval 2014 Laptop and Restaurant ABSA datasets (Pontiki et al., 2014). We chose this topic for our research since Aspect-based Sentiment Analysis can be a key indicator of satisfaction level of a user when looking at reviews or even tweets. The contribution of each member can be found in Appendix A

1 Analysis of the main idea

ABSA tasks involve detecting the sentiment from a text with regards to a specific aspect. Given the fact that this aspect constitutes the context of our prediction, one of the go-to models when having to identify contextualized information are transformers.

Similar in nature with a Recurrent Neural Network, they both process information using context. However, one key improvement in the architecture of the transformer is the encoder-decoder structure, which allows it to process the text without being restrained to the given order of sequences. This in turn means that it can analyze and detect the importance of each sequence in parallel.

One such transformer is BERT (Devlin et al., 2018), a bidirectional model, meaning that when compared to a classical transformer, it can train both ways. In order to retrain information with regards to the original position of tokens, the aforementioned transformers uses positional encoding. This type of training gives BERT the possibility of leaning context-dependent features for every word in the given text. The enforcing of this deep context learning is achieved by using Masked LM. The process consists of hiding at random some tokens and using them as target for predictions given the context tokens (words).

2 Related Research

In order to detect sentiments in a text when given a specific term or context, one must first extract meaningful information from the words in the form of word embeddings. One approach can be to use word representations that are not specific to the given task. As such, the 2 main approaches explored by researchers at the time consisted of using GloVe (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013) representations to produce the features needed for the models. They both rely for generating the embeddings on occurences of words in text. The difference in these 2 approaches consists in the fact that GloVe are trained on a much bigger corpus, while Word2Vec feature representations are obtained by using the frequency of occurance in the given dataset task.

The downside of this word embeddings generation method is that it is independent of the context. A similar approach with the one we proposed in this technical report has been evaluated in the paper "Exploiting BERT for End-to-End Aspect-based Sentiment Analysis" (Li et al., 2019). They use BERT as an embedding layer, meant to extract context-dependant representation for each word.

As such, each word is passed through the

BERT model by first generating the token embedding along with positional and segment embedding. They are then used as inputs through the transformer layers. Then the resulting hidden representations are fed into a separate layer, called E2E-ABSA layer. They propose several implementations for this layer, most notably a linear layer, a RNN, a self attention network or a Conditional Random Field. The ultimate goal of this layer is to predict the sentiment assigned for each token. They train and evaluate on the same datasets (Laptop and Restaurant from SemEval 2014).

The paper presents different results with each E2E-ABSA layer implementation and compares the effect of freezing all BERT layers in order to use more general representations with finetuning the components.

3 Approach and experiments

As previously mentioned, the first step in achieving the task of aspect-based sentiment analysis is generating embeddings from the given words (tokens).

For dataset, we are using Laptop and Restaurant from SemEval 2014. This data represents texts extracted from reviews of laptops and restaurants. Each review contains the actual text / message written by the user along with some aspect terms and the associated emotion. The sentiments belong to three main categories: positive, neutral and negative. For example, for the text "In the shop, these MacBooks are encased in a soft rubber enclosure - so you will never know about the razor edge until you buy it, get it home, break the seal and use it (very clever con)", two aspect terms are given: "rubber enclosure" which has a positive sentiment assigned to it and "edge" which, in this context, is negative.

For the preprocessing part, we read the text and split it into words. We go through each word and assign the corresponding sentiment class. As such, the token is given label O if the word is not among the given aspects, and the sentiment label for the corresponding aspect otherwise. As such, there are 4 main classes: O, T-POS, T-NEG, T-NEU. In implementation, we encode this labels to 0, 1, 2 and 3. If the sentence length is lower than 128 tokens, we pad it. Otherwise, we split it to make sure that the size is kept constant. One characteristic of BERT is that it splits tokens into subtokens. Since, as previously mentioned, the sentiments are specified at class level, we assign -100, a special label that we will skip when calculating the loss. We then store the data and labels inside a custom implementation of the Dataset class from PyTorch. This allows us to maintain a scalable format that can then be loaded inside a Dataloder for easy batch-based training and evaluation pipelines.

When looking at the dataset, one important thing to notice is the frequency of each class 4. As expected, label O is the dominant one, since the vast majority of words are not aspects with specified sentiments. In order to avoid the model getting heavily biased on this class, we apply 2 methods for assigning an importance factor (i.e. weights) to each class:

1. by dividing the number of elements from class O with the number of elements from a given class

2. by using compute_class_weight from sklearn

The training procedure utilized AdamW as the optimizer, incorporating dynamic learning rate adjustments through ReduceLROnPlateau. Training spanned approximately 15 epochs with Early Stopping implemented. Additionally, we saved the optimal model checkpoint based on the F1 micro score achieved on the evaluation set of the dataset.

We managed to train the models without frozen layers, yet we noted a variance between our results and those reported in the paper under the same setup. This difference can be observed in Table 1. Further investigation led us to identify two potential causes:

1. In their code, the training loss was calculated without assigning weights, despite the significant class imbalance in the dataset.

2. The evaluation code also included sub-tokens in the evaluation process. These sub-tokens are not intended for backpropagation or score calculation, as they solely represent parts of a main token.

Model	Datasets	
	Laptop	Restaurant
Our BERT + GRU	58.7	65.1
Paper BERT + GRU	61.1	70.2

Table 1: Comparison of our BERT results with those reported in the paper.

We trained each model head using identical configurations for both datasets. Our approach involved keeping the first 8 layers of the BERT backbone frozen while fine-tuning the remaining layers and the head. Specifically, we conducted training for 14 epochs on the laptop dataset and 9 epochs on the restaurant dataset, mirroring the training duration reported in the referenced paper. The results can be seen in Fig. 2.

Model	Datasets	
	Laptop	Restaurant
BERT + Linear	50.8	61.2
BERT + LSTM	49.8	61.0
BERT + GRU	53.1	64.4
BERT + TFM	45.5	60.5
BERT + SAN	48.0	58.8

Table 2: Comparison of partially frozen BERT model with various heads on two datasets.

We conducted training with the RoBERTa model as the backbone for the same heads; however, the results were inferior to those obtained with BERT. Despite using identical setups, the contrast between the models is illustrated in Table 3

For classification metrics, we started off with accuracy, which we defined as the number of correctly classified sentiments divided by the number of total samples. In order to better compare our results with already existing papers, we also implemented micro average F1 which first counts the number of TP, FN, FP.

Model	Datasets	
	Laptop	Restaurant
BERT + Linear	50.8	61.2
BERT + LSTM	49.8	61.0
BERT + GRU	53.1	64.4
RoBERTa + Linear	14.9	30.4
RoBERTa + LSTM	10.4	26.3
RoBERTa + GRU	19.1	35.6

Table 3: Comparing the partially frozen BERT model with the partially frozen RoBERTa.

As such, the metric represents the proportion of correctly classified observations out of all observations.

As it can be seen in Table 2, the best performing model for both datasets that we implemented was BERT with GRU layer. Looking closer at the confusion matrix 3 for laptop and 4 for restaurant dataset, we can see that the model is quite robust not only on class O, but also on sentiment classes, which indicates that the weighting mechanism is performing as expected. The evolution of the loss and performance metrics can be seen in 1 and 2. Similarly, both F1 macro and micro are detailed in 5 and 6.

4 Conclusions and Future Work

In the end, we experimented with multiple approaches to enhance the model. However, in conclusion, we believe our efforts primarily contributed to refining the evaluation process, leading a more correct model, although the accuracy was slightly lowered.

Future work could involve exploring additional techniques to further boost model performance, such as investigating new preprocessing methods or even employing ensemble learning strategies. Another way we could improve the results would be by conducting in-depth analysis on false positives/negatives i.e. wrong prediction patterns could provide insights for refining the model architecture or training process.

Dataset	Sentiment class	Encoded Class	Frequency	Manual weight	Sklearn weight
Laptop 2014	O	0	42080	1	0.268
Laptop 2014	T-POS	1	1222	34.435	9.228
Laptop 2014	T-NEG	2	1142	36.847	9.875
Laptop 2014	T-NEU	3	666	63.183	16.933
Restaurant 2014	O	0	37773	1	0.279
Restaurant 2014	T-POS	1	2818	13.404	3.750
Restaurant 2014	T-NEG	2	923	40.924	11.449
Restaurant 2014	T-NEU	3	759	49.766	13.923

Table 4: Train class distribution and counter-weights attributed

A Contributions

Aside from reading related research and understanding the models which were specific to both of us, here is a summary of contributions to the project of each member:

Constantin Gabriel-Adrian

1. elaborated the scripts for processing the datasets and plotting the results
2. implemented and experimented with Class-Weighted CrossEntropy, reduce learning rate on plateau and early stopping saving of the best model
3. implemented and experimented with unfrozen BERT model layers at different depths on Laptop and Restaurant SemEval 2014 Datasets
4. elaborated the introduction, related research, description of the approach and conclusion of the technical report

Sociu Daniel

1. implementation of BERT training and evaluation pipelines
2. adapted script to use different final layers on top of BERT and experimented with them
3. experimented with frozen BERT model and reproduced baselines from (Li et al., 2019)
4. elaborated the tested architectures, experiments and results sections of the technical report

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

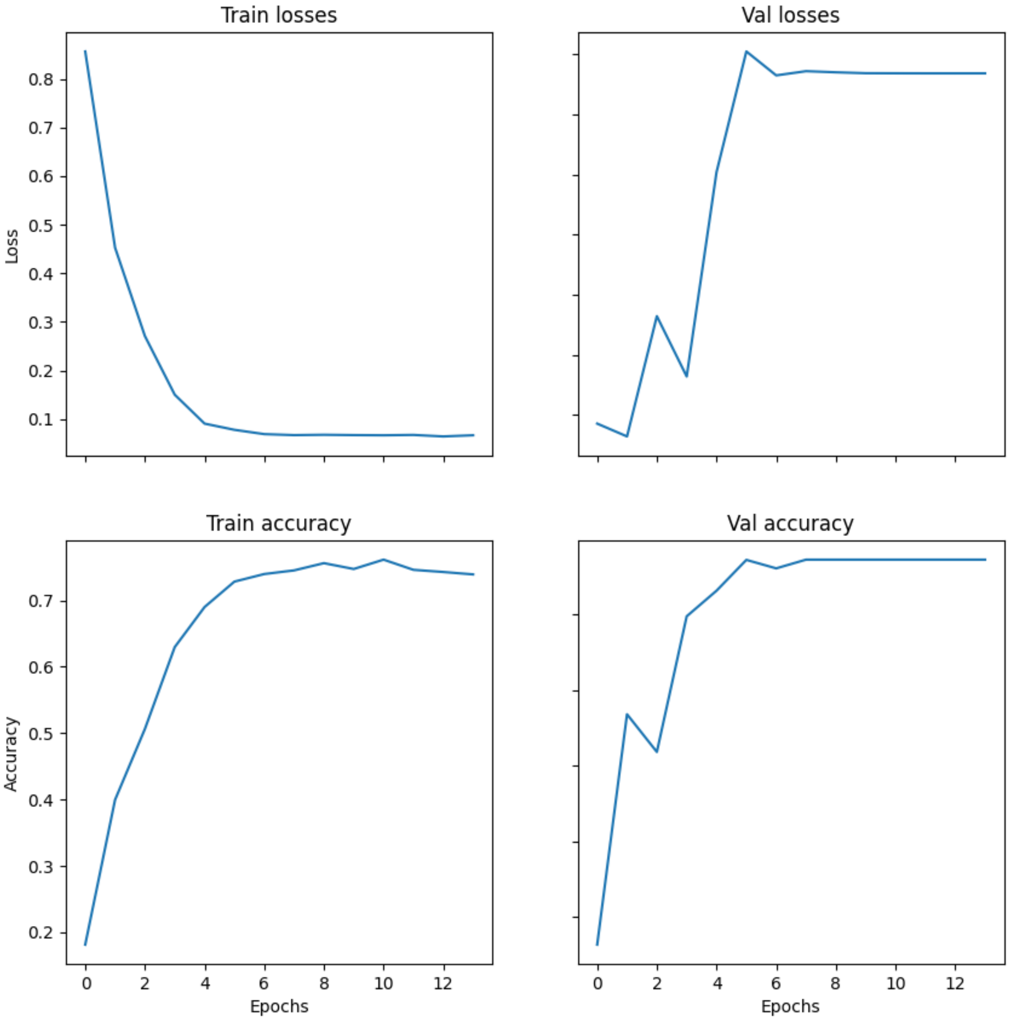

Figure 1: Loss and accuracy evolution on laptop

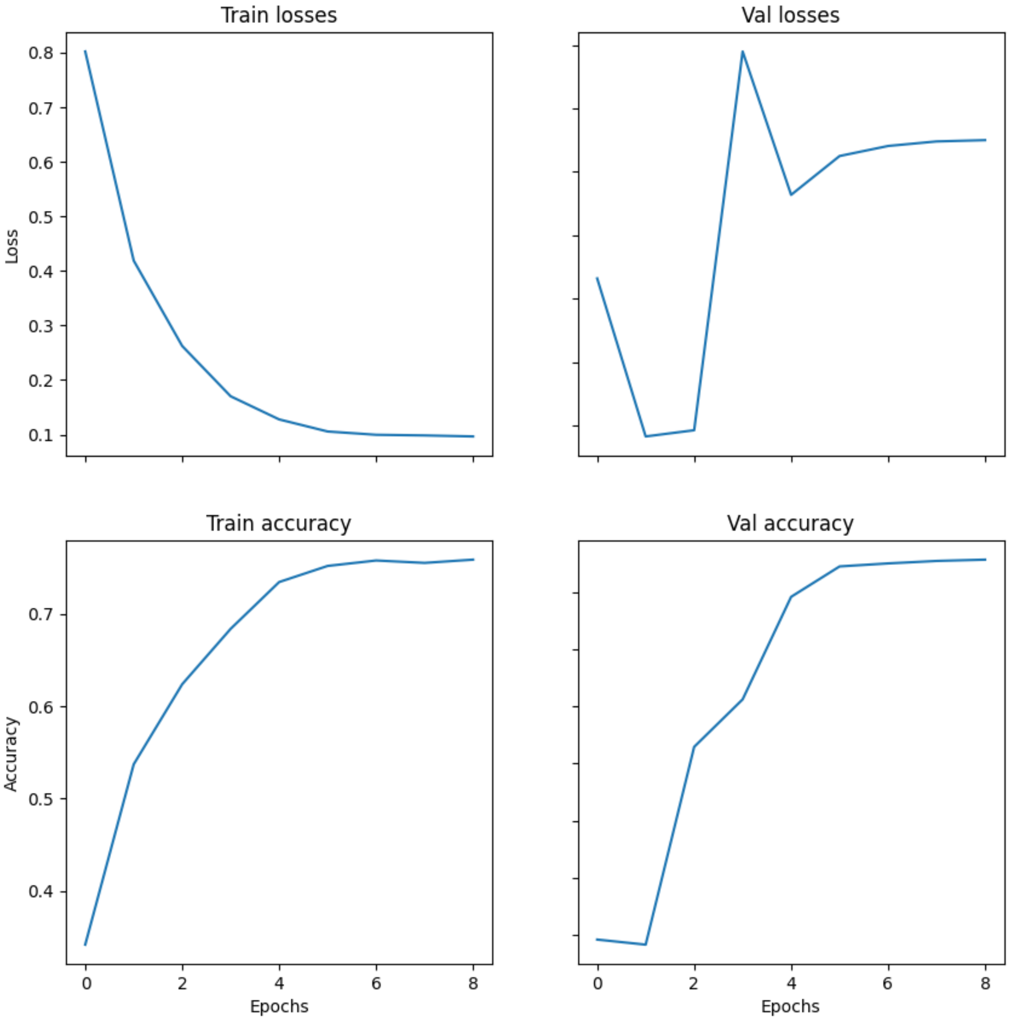

Figure 2: Loss and accuracy evolution on restaurant

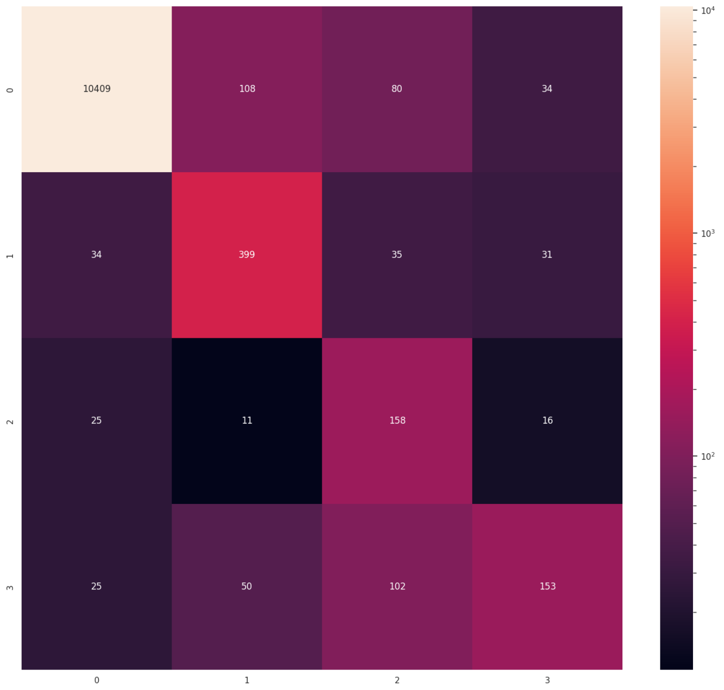

Figure 3: Bert confusion matrix on laptop

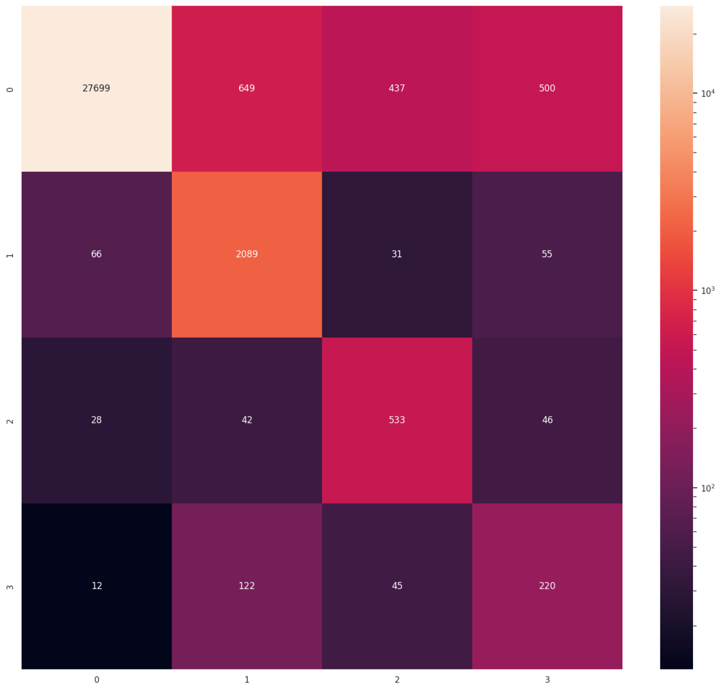

Figure 4: Bert confusion matrix on restaurant

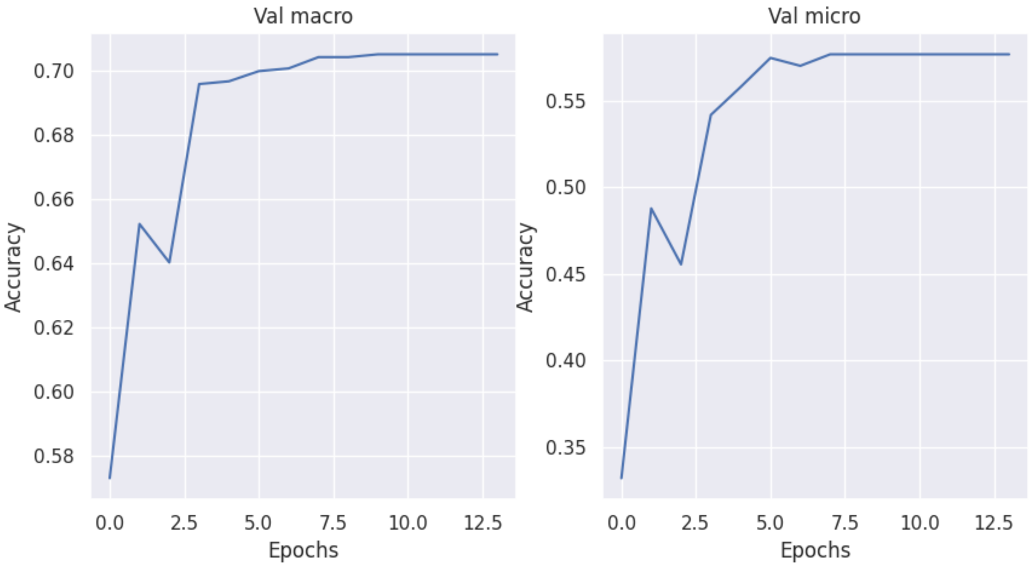

Figure 5: Bert F1 comparison on laptop

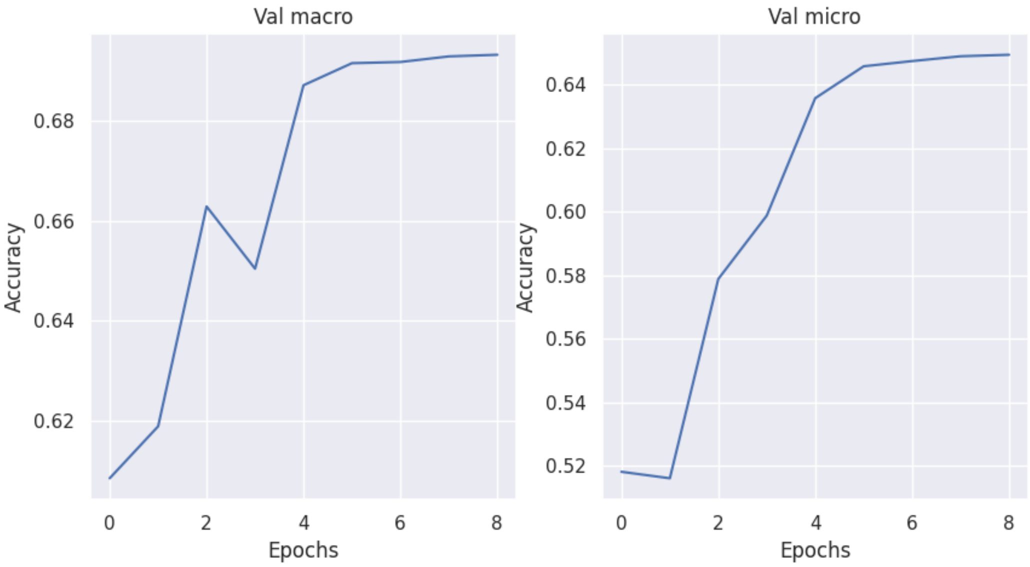

Figure 6: Bert F1 comparison on restaurant