Aspect-based Sentiment Analysis using Transformers

Constantin Gabriel-Adrian (507)
Sociu Daniel (507)

Problem definition

Task definition

- Aspect-based Sentiment Analysis:
  - involves predicting the sentiment of a specific paragraph when given a context / term
  - Can be a key indicator of satisfaction level of a user with regards to a specific aspect (for example when looking at reviews)

- Example:
  - 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)
  - Aspect-sentiment pairs:
    - "rubber enclosure" - positive
    - "edge" - negative

Dataset

SemEval 2014

The processed dataset turned out to have:

- 2700 training samples per dataset
- 800 test samples per dataset

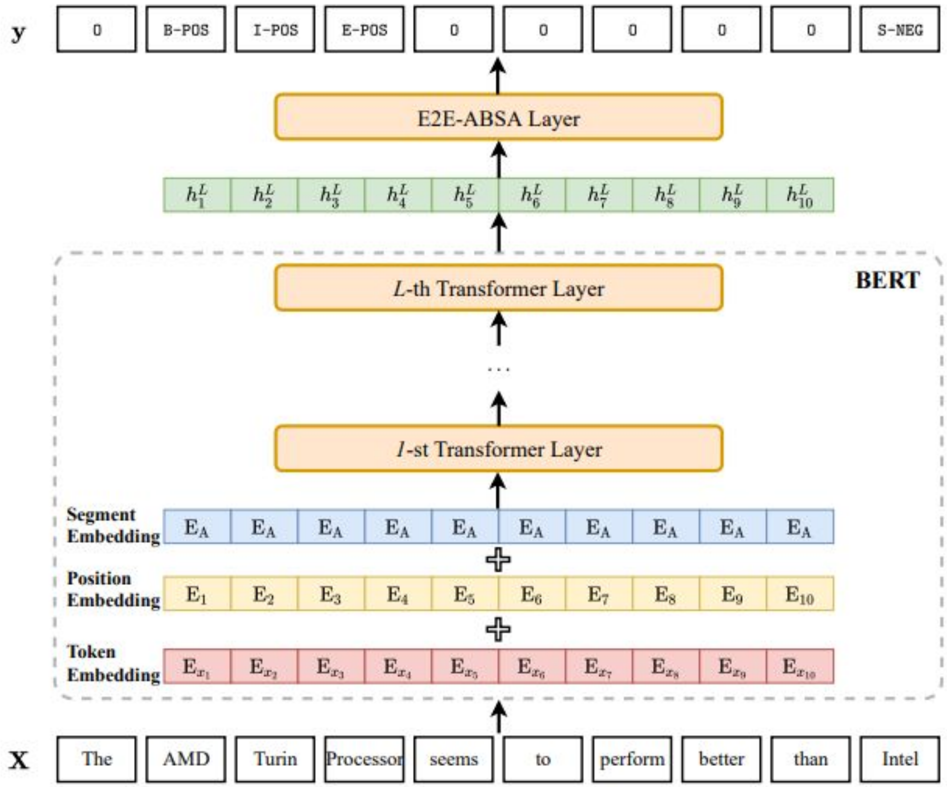
Dataset	Sentiment class	Encoded Class	Frequency
Laptop 2014	O	0	42080
Laptop 2014	T-POS	1	1222
Laptop 2014	T-NEG	2	1142
Laptop 2014	T-NEU	3	666
Restaurant 2014	O	0	37773
Restaurant 2014	T-POS	1	2818
Restaurant 2014	T-NEG	2	923
Restaurant 2014	T-NEU	3	759

Related research

Related Research

- Word embeddings generation
    - Context-independent:
        - GloVe
        - Word2Vec
    - Context-dependent:
        - RNNs
        - Transformers

BERT with different E2E-ABSA layers



Related Research - E2E-ABSA layer

- Proposed E2E-ABSA classifier layers
  - Linear Layer
  - LSTM
  - GRU
  - Transformer encoder layer
  - Self-Attention Networks
  - Conditional Random Fields
- These layers are trying to improve the predictions, given the transformer encodings as input

Proposed solution

E2E ABSA

- Write our own E2E python scripts using PyTorch in order to solve the problem
- Attempt to improve the results from the paper
- Keep the same layers from the original paper
  - Also test with HuggingFace linear implementation for token classification
- Try different backbone models:
  - BERT
  - RoBERTa

Experimental Setup

Training tricks & hyper-parameters

- AdamW as optimizer
- Dynamic learning rate changes via ReduceLROnPlateau
- About 15 epochs of training, with Early Stopping
- Weight-adjusted importance for each class
  - by dividing the number of elements from class O with the number of elements from a given class
  - by using compute_class_weight from sklearn
- Saved best model based on validation F1
  - Evaluated after each training epoch

Train class distribution and weights attributed

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

Metrics used

- Accuracy
  - Number of correctly classified sentiments divided by the number of total samples.
- Micro F1
  - Global average F1 which counts the number of TP, FN, FP
  - proportion of correctly classified observations out of all observations

$$
\begin{aligned}
\text{Micro F1 Score} &= \frac{\text{Net } TP}{\text{Net } TP + \frac{1}{2}(\text{Net } FP + \text{Net } FN)} \\
&= \frac{\sum_i^n M_{ii}}{\sum_{i=1}^n M_{ii} + \frac{1}{2}\left[(\sum_{i=1}^n \sum_{j=1 \text{ to } n;\, i \neq j} M_{ij}) + (\sum_{i=1}^n \sum_{j=1 \text{ to } n;\, i \neq j} M_{ji})\right]}
\end{aligned}
$$

Results

Results

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.

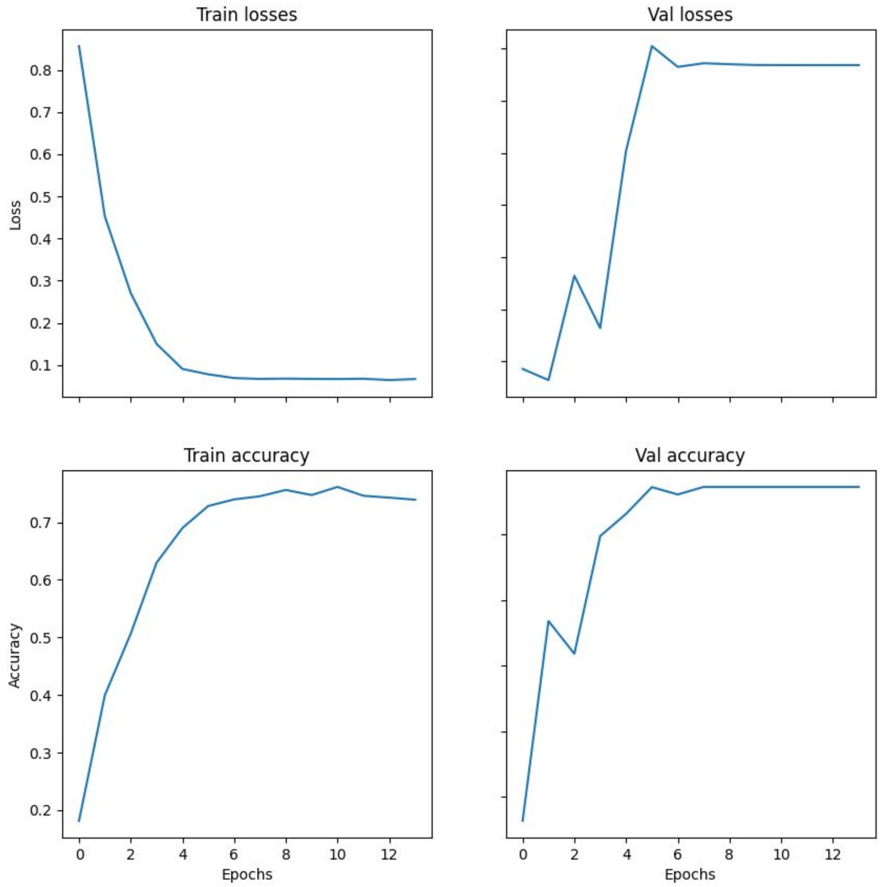
Model	Datasets	
	Laptop	Restaurant
BERT + Linear	50.8	61.2
BERT + LSTM	49.8	61.0
BERT + GRU	53.1	64.4
RoBERTa + Linear	14.9	30.4
RoBERTa + LSTM	10.4	26.3
RoBERTa + GRU	19.1	35.6

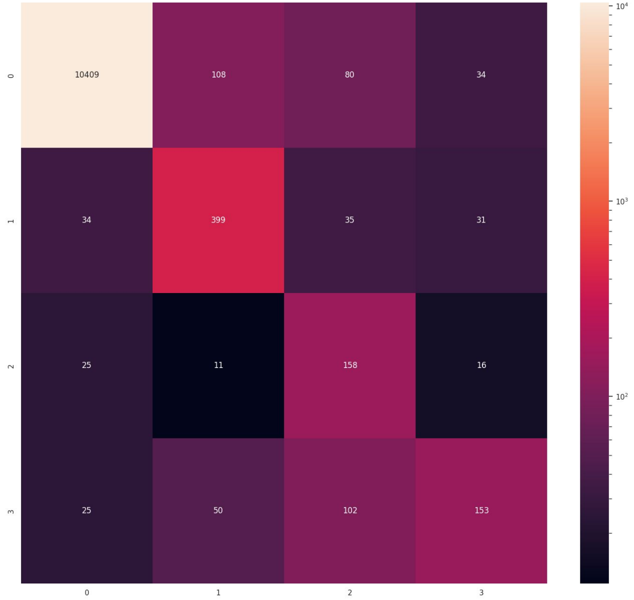
Table 3: Comparing the partially frozen BERT model with the partially frozen RoBERTa.

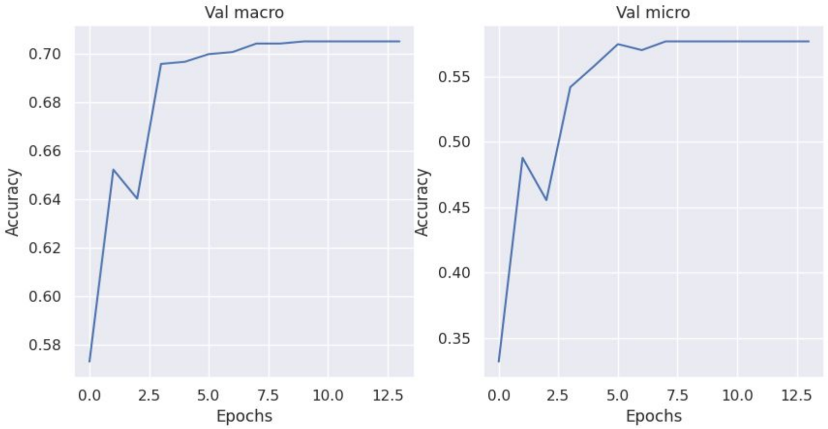
Model	Datasets	
	Laptop	Restaurant
BERT + Linear	50.8	61.2
BERT + LSTM	49.8	61.0
BERT + GRU	53.1	64.4
BERT + TFM	45.5	60.5
BERT + SAN	48.0	58.8

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

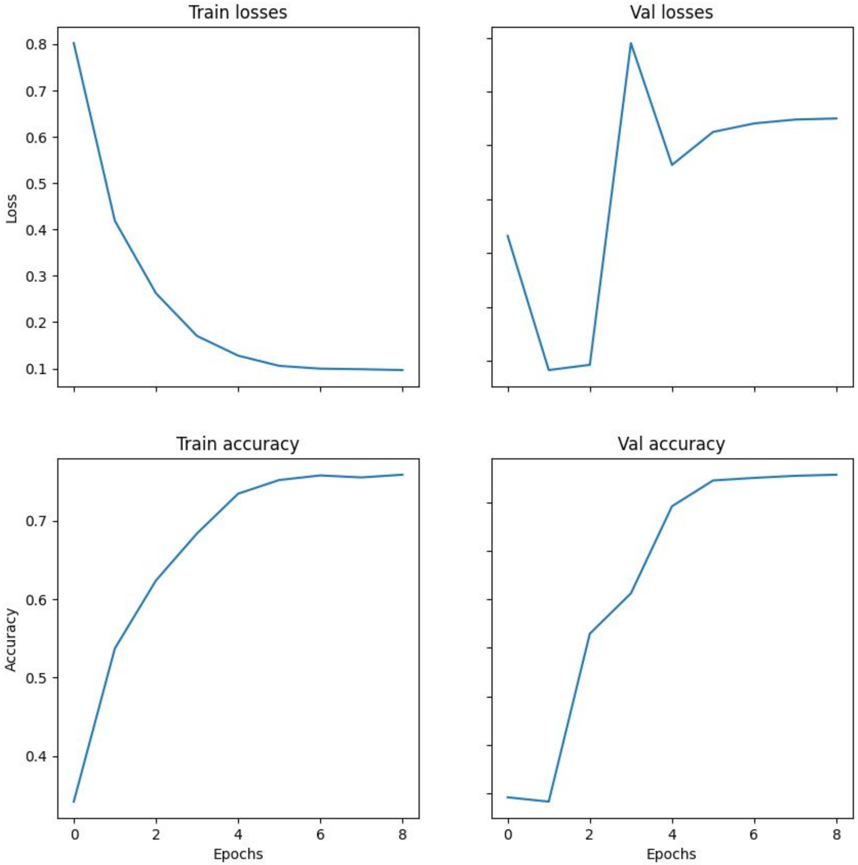
Plots - best model - laptop

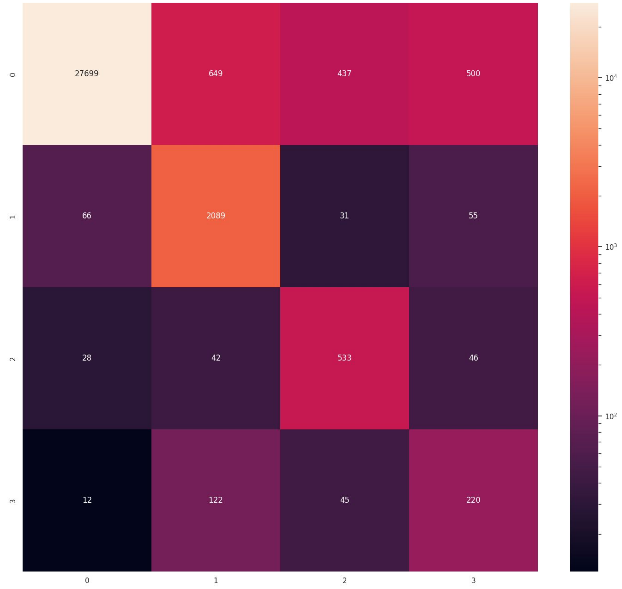


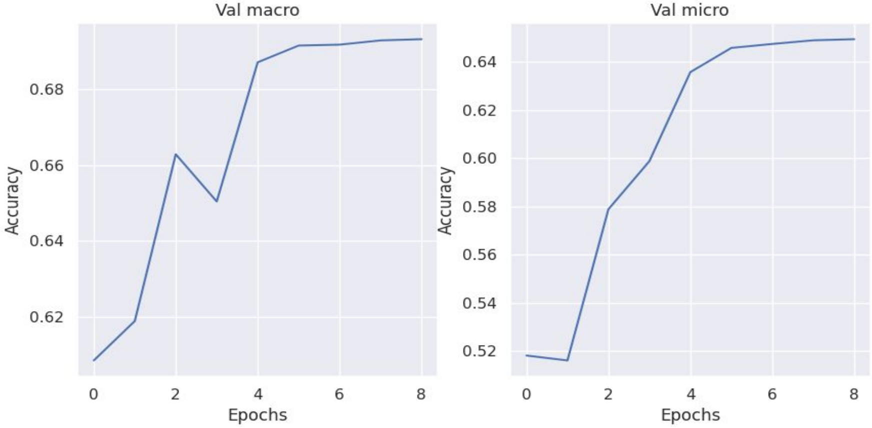




Plots - best model - restaurant







Conclusions and Future Work

Conclusions

- Improvements in the pipeline:
  - Added weighted training.
  - Speed-up training with frozen layers while keeping good accuracy.
- Evaluation:
  - They were evaluating on the split tokens of each word
  - They considered these tokens class “O”
  - We filtered all those tokens out.
- Although, those didn’t bring any improvement to the results.

Thank you!