Aspect-Based Sentiment Analysis

Lorenzo Proietti 1754547
Sapienza University of Rome
proietti.1754547@studenti.uniroma1.it

1 Introduction

This work is composed by two tasks. The first task is called *aspect term identification* where, given a sentence, we have to identify the aspect terms in it, that correspond to words towards which there is a sentiment in the sentence. The second task is called *aspect term polarity classification* in which, given the previously extracted aspect terms and the input sentence, we have to identify the sentiment associated to them among four possible classes: *positive, negative, neutral* and *conflict*.

2 Preprocessing

The only preprocessing done on the input sentences was to remove the punctuation.

3 Models

3.1 Aspect term identification model

Aspect term identification can also be seen as a named entity recognition task. In this scenario, I decided to classify each word in a sentence with two possible labels: O or AT, where O indicates that the word is not part of an aspect term, while AT indicates the opposite. Consecutive words with associated label AT are joined together to form a single aspect term. This model has been implemented with different configurations of *BERT* (Devlin et al., 2019) and, the one with which I get the best results, is *RoBERTa large* (Liu et al., 2019). The architecture of this model is the same regardless of the Bert configuration used and I decided to apply a fine-tuning strategy. In particular, I concatenated the last four hidden layers of the Bert model, as suggested in its paper and, in order to obtain a single contextualized representation for each single word in the input sentence, I decided to combine the results for all word pieces as represented in figure 1, taking inspiration from (Navigli, 2021), where the fully connected layer contains a number of units that is equal to the hidden size of the Bert model used. Then, the output of the previously described architecture is passed to another fully connected layer with a number of units equal to half the hidden size of the Bert model used and with a ReLU activation function, which output is passed to the final dense classifier, where a Cross Entropy Loss function was used. The hyperparameters settings can be seen in table 1 (values suggested in the Bert paper were taken into consideration). A schedule with a learning rate that decreases linearly from the initial value to zero was used. The model selected was the one that achieved the lowest loss during the four training epochs. With these settings I tried four bert models: Bert base, Bert large, Roberta base and Roberta large and, in all of them, I considered a cased tokenization, since it is more suitable for a named entity recognition task. The results from these four models can be seen in table 3.

3.2 Aspect term polarity classification model

The input of this model is composed by a sentence and the aspect terms in it and, the corresponding outputs, are the polarities associated to the given aspect terms in the input sentence. Since also in this case we have a classification task, the Cross Entropy Loss function was used but, considering that the labels distribution in the training and validation set is highly unbalanced (as can be seen in tables 2 and 4), I decided to give weights to the classes in the loss computation, since in this way, when the optimization algorithm tries to optimize the parameters of the model considering the loss value, higher weights to less frequent classes help the model to understand that it must optimize the parameters with respect to these classes with the same measure of the others, otherwise a model that never returns the conflict class would have a good loss, but a low macro F1 score. So, the loss of a

given sample (with respect to a class) and the mean loss of a batch of N elements will be:

$$loss(x, class) = weight[class] \cdot \cdot \left(-x[class] + \log \left(\sum_j \exp(x[j]) \right) \right), \quad (1)$$

$$loss = \frac{\sum_{i=1}^{N} loss(i, class[i])}{\sum_{i=1}^{N} weight[class[i]]} \quad (2)$$

where $x[j]$ is the logit value of the class j. The weight given to the class c_i is equal to $\sqrt{N_c/n_i}$, where N_c is the number of occurences of the most frequent label in the training set (in this case, positive class, that has 2605 samples) and n_i is the number of occurences of the class c_i. I decided to consider the square root since with it I obtain better results (as shown in in Aspect term polarity classification model variants section), probably because without it, the weight given to classes such as *conflict* is too high. Also in this model, Roberta large with a cased tokenization was used and, in order to classify the polarity of a given aspect term in the input sentece, I combine the contextualized embeddings of the word pieces of the aspect term through a self-attention mechanism (Vaswani et al., 2017). In particular, given v_i the last hidden layer of Roberta large of the i-th word piece of an aspect term, its attention score is computed as $s_i = a^T v_i$, where a is a vector that is trained. At this point, the representation of an aspect term that is composed by n word pieces is obtained as:

$$\sum_{i=1}^{n} s_i v_i \quad (3)$$

Now, this representation is passed through a fully connected layer with a number of units equal to the hidden size of the Roberta model used and with a Tanh activation function and, finally, the output of this fully connected layer is passed to the dense classifier layer, where the weighted Cross Entropy Loss function was used. The architecture of this model is summarized in figure 2. The hyperparameters settings can be seen in table 6 (values suggested in the Bert paper were taken into consideration). A schedule with a learning rate that decreases linearly from the initial value to zero was used.

4 Overall results

The two models previously described were trained separately on the full training set of reviews of both laptops and restaurants, using a Tesla T4 on Google Colaboratory. After that, they were combined sequentially, that is, given a sentence in input, first the aspect terms in it were extracted using the first model and, considering as subsequent inputs the sentence itself and the extracted aspect terms (which can also be a empty set), the second model will classify the polarity associated with both. The results of this approach, considering the two submitted models, are shown in table 8.

5 Extras

5.1 Aspect term identification model variants

The variants for this model I tested are characterized by a different output that is taken from the Bert model. In particular, the alternatives mentioned in the Bert paper were taken into consideration and, with the same hyperparameters and rest of the model architecture, the results of these variants can be seen in table 5. The variants were tested only with the best model, so Roberta large. Technically, the model with the highest F1 score is the one that only considers the last layer of the Bert model, but with this strategy I get a higher loss (0.0618) than the one I obtain with the model that takes into account the concatenation of the last 4 layers (suggested by Bert paper), that is 0.0546, so for this reason I decided to submit the latter. Another variant implemented for this model makes use instead of a BiLSTM (Hochreiter and Schmidhuber, 1997) and Conditional Random Field (CRF) (Lafferty et al., 2001), which is a method thanks to which a prediction of a determined NE label can be carried out also taking into consideration the previous tags of a given sequence. To do this, a graphical model is considered for the predictions. In our case, linear CRF is used, in order to consider linear dependencies between predictions. However, having only two labels, this method should not bring great advantages but, the main drawback, is the use of a BiLSTM to do sequence tagging, instead of a Bert model. In fact, as can been seen from table 5, the results obtained with this method are by far the worst. The BiLSTM method used the *GloVe 840B 300d* pretrained vectors (Pennington et al., 2014).

5.2 Aspect term polarity classification model variants

As variants for this model I have also considered here different outputs taken from Roberta large, but also different weighting schemes for the loss. In particular, I considered the same scheme previously defined, but without applying a square root to all weights. I also considered the so called Inverse of Number of Samples (INS), where the weight of the c_i class is equal to $1/n_i$ and also its square rooted version Inverse of Square Root of Number of Samples (ISNS). Another tested weighing scheme was the Effective Number of Samples (ENS) (Cui et al., 2019), which authors claim to associate with each sample a small neighboring region rather than a single point. The effective number of samples is defined as the volume of samples. In this scheme the weights are defined as follows:

$$w_c = \frac{1}{E_{n_c}} \tag{4}$$

$$E_{n_c} = \frac{1 - \beta^{n_c}}{1 - \beta} \tag{5}$$

where n_c is the number of samples in class c, E_{n_c} represents the Effective Number of Samples and $\beta \in [0, 1)$ is a hyperparameter. As suggested in the paper, I chose $\beta = 0.9999$. The results of the overall comparison can be seen in table 7, where the best model described in Aspect term polarity classification model section is named as ATPC with sqrt.

Hyperparameter	Value
Batch size	32
Dropout	0.1
Optimizer	AdamW
Learning rate	3e-5
Weight decay rate	0.01
Number of epochs	4
Maximum gradient norm	1.0

Table 1: Aspect term identification model hyperparameters.

Polarity label	Number of occurrences
positive	2605
negative	1364
neutral	877
conflict	111

Table 2: Labels distribution in aspect term polarity classification task training set.

Model	Precision	Recall	F1
BERT base	74.50	79.94	77.13
BERT large	79.58	81.23	79.37
RoBERTa base	80.17	79.94	80.06
RoBERTa large	**80.91**	**84.20**	**82.52**

Table 3: Aspect term identification models results, the best scores are in bold.

Polarity label	Number of occurrences
positive	546
negative	307
neutral	216
conflict	25

Table 4: Labels distribution in aspect term polarity classification task validation set.

Variant	Precision	Recall	F1
Second-to-Last Hidden	81.10	82.90	81.99
Last Hidden	**83.57**	82.72	**83.14**
Average Last Four Hidden	78.67	82.16	80.38
BiLSTM + CRF	72.69	69.44	72.25
Concat Last Four Hidden (submitted)	80.91	**84.20**	82.52

Table 5: Aspect term identification model variants, the best scores are in bold.

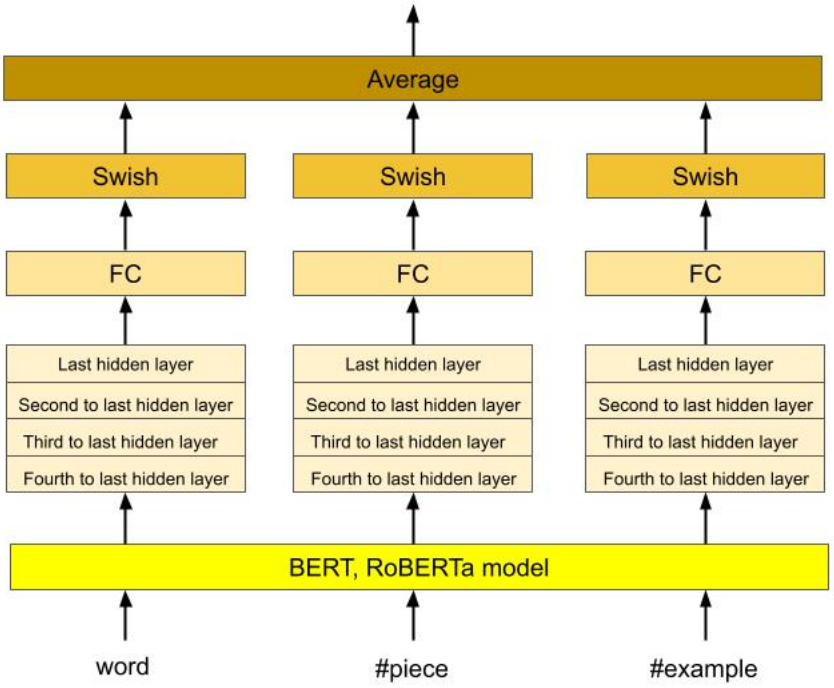

Figure 1: Aspect term identification model word pieces contextualized embeddings aggregation.

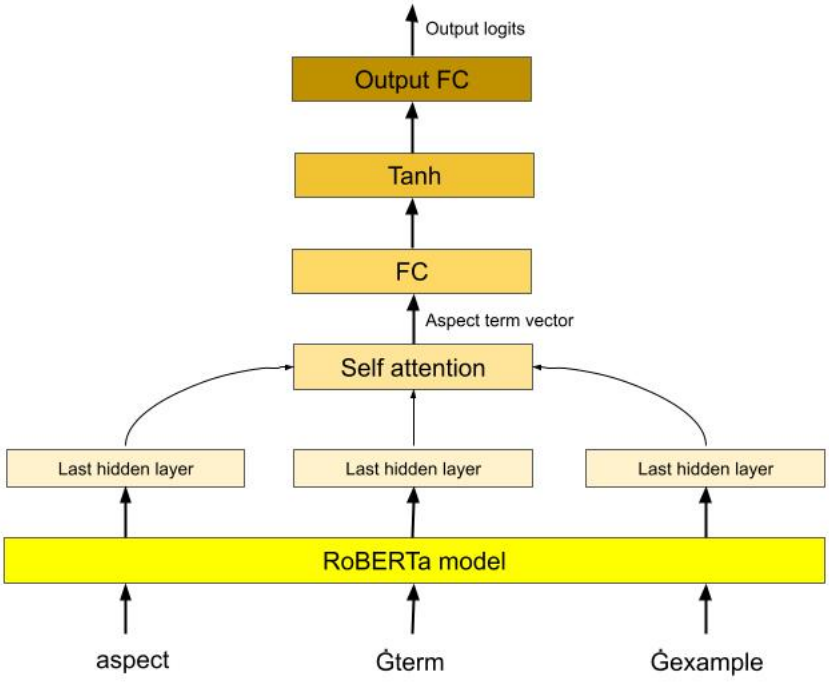

Figure 2: Aspect term polarity classification model architecture.

Hyperparameter	Value
Batch size	32
Dropout	0.1
Optimizer	AdamW
Learning rate	3e-5
Weight decay rate	0.01
Number of epochs	3
Maximum gradient norm	1.0

Table 6: Aspect term polarity classification model hyperparameters.

Variant	Micro F1	Macro F1
Average Last Four Hidden	**79.50**	63.23
Sum Last Four Hidden	79.13	60.12
INS	79.10	65.28
ISNS	77.60	64.45
ENS	75.80	61.87
ATPC	78.91	62.88
ATPC with sqrt (submitted)	**79.50**	**66.95**

Table 7: Aspect term polarity classification model variants, the best scores are in bold.

Model	Micro F1	Macro F1
Concat Last Four Hidden (submitted) + ATPC (submitted)	**65.70**	**55.69**

Table 8: Overall results.

References

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. volume 9, page 1735–1780, Cambridge, MA, USA. MIT Press.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Roberto Navigli. 2021. Scaling semantic role labeling and semantic parsing across languages. Leibniz Universität Hannover LUH; L3S Research Center; CLEOPATRA ITN. Https://doi.org/10.5446/52947 $Lastaccessed$: $11Jun2021$.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.