NLP Homework2-2021: Aspect-Based Sentiment Analysis

Sorokoletova Olga
sorokoletova.1937430@studenti.uniroma1.it

July 13, 2021

1 Introduction

The aim of the homework is to solve ABSA task at the sentence level. All 4 models as separate units (a - Aspect Extraction, b - Aspect Sentiment, c - Category Extraction, d - Category Sentiment) as well as in form of join units (a+b, c+d) have been implemented and evaluated on the given datasets. Regarding a+b and c+d, they are implemented in two possible ways both: sequentially (a is followed by b, c is followed by d) and as a combined model solves two tasks simultaneously. The overall idea of a proposed solution is to treat the problem as a sentence pair classification and use the pre-trained language model BERT (and its modifications) for the contextual word representations. To represent the power of the transformer models in comparison with more old-fashioned approaches, naive baseline model that uses not contextual word representations was implemented for the task a.

2 Pre-processing

Pre-trained BERT tokenizers which make use of so-called **wordpiece tokenization** were mainly used in the transformers models. For the naive baseline – word tokenizer, provided by **nltk** + punctuation removal, but since punctuation removal led to the misclassification in some particular cases, finally it was cancelled.

The task assignment in the form of sentence pair classification (determining the semantic relations between two sentences) means that BERT is used to compare two sets of sentences: the first one – real sentences provided in the datasets, and the second one – aspect or category terms. With this purpose BERT tokenizer provides two special tokens: $< cls >$ – to denote the beginning of the sentence and $< sep >$ – separation token. As for the naive model, $< unk >$ and $< pad >$ tokens are added manually to pad the sequence and handle out of vocabulary words.

In both cases, further step is to process tokens through embeddings. Three embedding layers: Token Embedding Layer, Segment Embedding Layer and Position Embedding Layer in **BERT are used as pre-trained**, and option to choose dataset-based or pre-trained **GloVe** embeddings is left up to user in naive model (the best performance achieved with Glove).

3 Experiments

3.1 Aspect Extraction: Naive vs BERT

Naive models uses $0/1$ flags to denote desired aspects in the training examples and maps them to the input tokens. Architecture is represented by one lstm layer that is followed by the classification layer and allows to choose if one would like or not make use of the pre-trained embeddings. Hyperparameters set up for this model can be found in Table 1. And Table 2 displays comparison between best obtained performance of Naive model and best performance for Aspect Extraction obtained with BERT model.

Hyperparameter	**Value**
Epochs	100
Batch Size	128
Embedding dim	100
Embedding	GloVe
N hidden units	128
Window Size	100
Window Shift	100
Optimizer	Adam
Learning Rate	10^{-4}
Dropout rate	0.0

Table 1: Hyperparameters corresponding to the best performed Naive model.

Model	F1
Naive	67
DistilBERT	83

Table 2: Best performance of aspect extraction classifiers on the restaurants dataset.

3.2 Choose best BERT

BERT, Small BERT, DistilBERT and Large BERT are tried as pre-trained for all set of tasks. The Table 3 allows to see hyperparameters chosen for each model and their performance on Aspect Extraction for the restaurants dataset. It can be concluded, that Distil BERT and Large BERT perform on the same level and better than other models. Therefore, Distil BERT is chosen as the best, because it executes faster.

However, for Aspect Sentiment task (and therefore, for Aspect Extraction + Aspect Sentiment too) as well as for Category Extraction, Category Sentiment and Category Extraction + Category Sentiment Large BERT tends to perform better (see the Table 4).

Model	Epoch	Batch Size	lr	F1
Small BERT	10	24	$2e{-}5$	79
BERT	4	16	$2e{-}5$	81
DistilBert	4	24	$2e{-}5$	83
Large BERT	4	24	$2e{-}5$	83

Table 3: Comparison of different BERT models for Aspect Extraction on the restaurant dataset.

Task	DistilBERT	Large BERT
b	55	60
a+b	44	48
c+d	48	56

Table 4: Comparison of F1-macro of Distil BERT and Large BERT models for Aspect Sentiment (b), Aspect Extraction + Aspect Sentiment (a+b) and Category Extraction + Category Sentiment (c+d) tasks on the restaurant dataset.

3.3 Combined model

Combined model is an alternative of a sequential execution of models a and b or c and d to solve task a+b or c+d. Its implementation is described in details in the notebook, devoted to the Category Extraction + Category Sentiment. It treats n-classification problem with n – number sentiments as $n+1$-classification problem adding class ”none” to denote that given aspect/category is unrelated to the given sentence.

For both a+b and c+d both sequential and combined approached were tried, and result of the comparison for the restaurant dataset are represented in the Table 5. Sequential approach performs better on aspects, meanwhile for the category detection and classification, combined approach is more suitable. Probably, the reason is that combined a+b tends to take into consideration too much of non-informative tokens, and then more smart manual and data-adapted way of pre-processing could help.

Additionally, combined models allow to estimate each of the sub-tasks separately. Therefore, we need to check the hypothesis if a,b,c,d obtained from the corresponding combined models perform better than each of them separately. Results (again for the restaurants) are shown in the 6. It’s clearly seen that hypothesis is wrong in all 4 cases at least for this dataset.

Task	Sequential	Combined
a+b	48	40
c+d	50	56

Table 5: Comparison of F1-macro of Sequential and Combined models for Aspect Extraction + Aspect Sentiment (a+b) and Category Extraction + Category Sentiment (c+d) tasks on the restaurant dataset.

Task	As is	From Combined
a	83	82
b	60	50
c	82	82
d	64	55

Table 6: Comparison of F1-macro for Aspect Extraction (a), Aspect Sentiment (b), Category Extraction (c) and Category Sentiment (d) tasks on the restaurant dataset obtained from each of the models separately and estimated from combined model.

3.4 Aspect Sentiment: Restaurants vs Laptops vs mixed

The most of the conclusions made so far are made based on the restaurants dataset. Meanwhile, all listed experiments were performed for the laptops dataset and mixed dataset as well. For example, in the Table 7 we can take a look on the macro-F1 performance of the best performing model for Aspect Sentiment task on all dataset. In general, laptops seem to be slightly more difficult dataset. Note: in the reference table the results are reported for the models that were trained and evaluated inside the same domain (train on restaurants – evaluate on restaurants, train on laptops – evaluate on laptops, train on mixed – evaluate on mixed), meanwhile cross-domain evaluations were also done, but for the mixed dataset on which overall evaluation of Aspect Sentiment Classification task is going to be performed, training not on the only one domain doesn't make sense, especially considering that sizes of both datasets are small.

Dataset	F1
RR	60
LL	54
MM	59

Table 7: Comparison of the best F1-macro for Aspect Sentiment tasks on the different datasets: R – restaurants, L – laptops, M – restaurants and laptops mixed, the first letter stands for training dataset, the second one – for validation.

3.5 Confusions

Finally, we can draw confusion matrices to detect the most problematic classes. Working on aspects, we will do it for the mixed dataset, and working on categories – only for restaurants.

First, let's plot confusion matrix for each of models a,b,c,d separately in the Figure 1. Looking at them, it seems that both binary classification problems are solved finely, but we have to remember, that confusion matrices are calculated on the row predictions, so in the case of Aspect Extraction we calculate confusion matrix based on the predictions on tokens, i.e. words, and not aspect terms, and then there is also decoding part that transforms predictions on tokens into predictions on aspect terms, and it's more correct to make conclusions based on them. Now, regarding Sentiment Classification part, the most hardly recognizable class is ¡¡conflict¿¿, as it was expected, because this is one of the most challenging parts of the Sentiment Analysis in general, and model should capture the context really well to be successful in handling it. The second-rate in ranking of hardly recognizable classes is ¡¡neutral¿¿, and it seems to be a matter of unbalancing in datasets, i.e. probably, model haven't seen enough examples of this class.

Next we can plot confusion matrices also for the combined forms of a+b and c+d models in the 2. Combined models tend to be shifted to the ¡¡none¿¿ class due to the fact that we artificially created the branching by adding a lot of training examples of this class, in particular, in the aspect-based part each training example was spread by the factor of number of words in the sentence. Apart from that, tendencies remains the same as in the case of separate models, but there is one interesting moment: a+b confusion matrix contains highlighted area of ¡¡positive¿¿ examples which were predicted as either ¡¡conflict¿¿ or ¡¡neutral¿¿ ones, meanwhile c+d matrix doesn't have this tendency. From the another side, it could be provoked by the fact that a+b model is overall weaker.

4 Conclusion

Aspect-Based Sentiment Analysis is a challenging task, but we can achieve decent performances by treating it as a sentence pair classification problem and applying pre-trained language transformer model from a BERT family of models to capture the context in the best possible way. The following list of ideas could help to get better results: adding of a more training data, upgrading of utility function for decoding predictions into aspect terms, usage of transformers that were trained exactly on the ABSA task, retraining of some layers after loading pre-trained transformer and more precise fine-tuning, working on the ¡¡conflict¿¿ class examples.

5 Extras

Model c+d doesn't reach macro-F1 equal to 75, but implemented and even with some additions (c as separate, d as separate, c+d as sequential). Apart from that: 1) a+b is also implemented in two forms, 2) baseline model is implemented for the convenient comparison of what we could do without transformers and what we can do now, 3) qualitative comparisons and plots are represented.

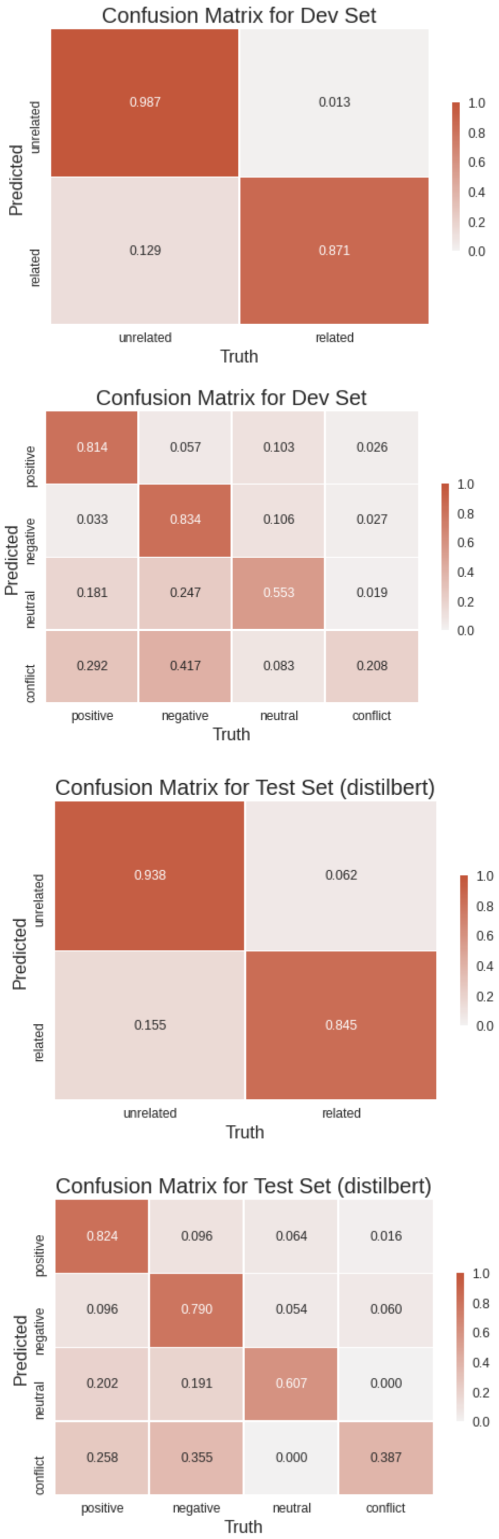

Figure 1: Aspect Extraction (1), Aspect Sentiment (2), Category Extraction (3), Category Sentiments (4) separately estimated on the mixed/restaurants dataset.

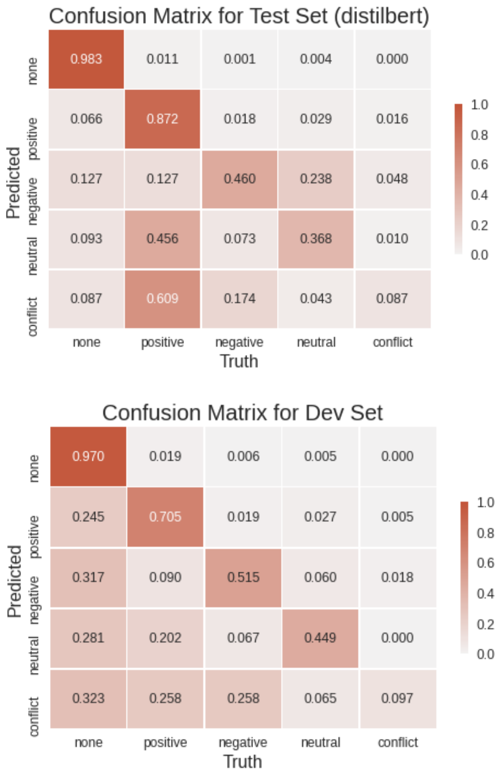

Figure 2: Aspect Extraction + Aspect Sentiment (1), Category Extraction + Category Sentiments (2) combined models on the mixed/restaurants dataset.