# NLU course projects - Assignment 3 - SA

*Davide Cavicchini (247823)*

University of Trento

davide.cavicchini@studenti.unitn.it

## 1. Introduction

For this assignment, we are requested to Implement a model based on a Pre-trained Transformer model (such as BERT or RoBERTa) for the Aspect-based Sentiment Analysis task regarding the extraction of the aspect terms only. The dataset used is the Laptop partition of the SemEval2014 task 4.

## 2. Implementation details

The implementation used is very similar to the one for the second part of the second assignment. The main difference is in how the data is prepared and evaluated. To prepare the data I map the labels into the reduced set [”T”, ”O”], and then collect the words as tokenized by the dataset.

At first, I tried to feed the transformer encoder with the pure sentence, but this resulted in many edge cases when selecting the first token for each word as considered by the dataset. The main problems were ignored punctuations, and the separation of abbreviated words, like ”it’s” tokenized as [”it”, ”’s”] by the dataset, especially when in the real sentence they do not include the apostrophe like in ”dont” which is mapped into [”do”, ”n’t”] regardless.

For this reason, I decided to simplify the problem and cast it as word-level tokenization by taking the single tokens considered by the dataset and concatenating them separated by a space. So, for example, a resulting sentence would be ”I do n’t like this computer , it ’s slow .”, which I can process the same way I did for the second part of the second assignment, and feed the classifiers using the first token representations following the idea from this paper [1].

As per the other assignments, I also opted to test out the effect of a dropout layer right after the transformer encoder for the same reason I explained in the previous assignment report.

The loss is a simple cross-entropy with weights giving the same importance to each class in a batch. As optimizer, I decided to use AdamW with a learning rate of $1e-4$.

## 3. Results

The requested metrics to evaluate the model results are the F1 score, the precision, and the recall. In Table 1 are presented the results from 5 training runs over the training set for the base models.

As for the previous assignments, the dropout allows the models to achieve higher scores across the board.

The best encoder model seems to be RoBERTa base, which outperforms in its basic training configuration even the BERT models trained with dropout.

| Model | **F1** ($\sigma$) | **Precision** ($\sigma$) | **Recall** ($\sigma$) |
|---|---|---|---|
| bert-base-uncased | 97.98 (0.10) | 97.98 (0.10) | 97.98 (0.10) |
| bert-base-cased | 97.76 (0.10) | 97.76 (0.10) | 97.76 (0.10) |
| roberta-base | **98.25 (0.13)** | **98.26 (0.13)** | **98.26 (0.13)** |
| bert-base-uncased Drop | 97.95 (0.11) | 97.96 (0.11) | 97.96 (0.11) |
| bert-base-cased Drop | 97.87 (0.05) | 97.88 (0.05) | 97.88 (0.05) |
| roberta-base Drop | <u>**98.27 (0.02)**</u> | <u>**98.28 (0.02)**</u> | <u>**98.28 (0.02)**</u> |

Table 1: *Results with the std scores for each configuration.* ***bold*** *values correspond to the best performance for that training setup, and <u>underlined</u> values the best overall.*

## 4. References

[1] Q. Chen, Z. Zhuo, and W. Wang, “Bert for joint intent classification and slot filling,” 2019. [Online]. Available: https://arxiv.org/abs/1902.10909