NLU course projects: SA

Rossana Cervesato (mat. 231499)

University of Trento

rossana.cervesato@studenti.unitn.it

1. Introduction

This is the final report for the Natural Language Understanding course, lab 6 (A.Y.2023/2024). The goal of the project is to implement a model based on BERT pre-trained language model for the Aspect-Based Sentiment Analysis (ABSA) task, specifically focused on extracting aspect terms.
This project includes:

- the analysis of the the Laptop partition of SemEval2014 task 4 dataset [1] (Table 1) (1);
- the implementation of the proposed model based on BERT;
- the proposed model's fine-tuning and evaluation.

The best model achieved a F1 score of 0.64, a precision score of 0.76 and a recall score of 0.55. The code is available at https://github.com/rossana24/NLU2024

2. Implementation details

2.1. Architecture

The proposed model architecture uses a pre-trained BERT-base model [2, 3] as its core component for encoding input text sequences. One linear classifier is applied on top of the BERT outputs to predict the start and end logits of the aspect spans. A custom distant cross-entropy loss (2) is utilized to compute the loss for start and end position predictions, with optional masking to handle invalid positions.

2.2. Metrics and Evaluation

The models' performance was evaluated using three metrics [4]: F1-score (3), precision (4) and recall (5).

This project primarily follows the methodology outlined in the paper by Minghao Hu et al. [3]. However, to incorporate the evaluation functions proposed by [4], the evaluation process involves converting model logits into BIO tags and then into BIOES tags to assess performance using precision, recall, and F1 score metrics. These conversions are consistent and should not introduce variability, ensuring that the evaluation outcomes remain reliable and aligned with the objectives of the original approach.

During the evaluation step, the trained model generates start and end logits for each token in a sequence. These logits represent the raw scores predicting the likelihood of each token being the start or end of an aspect term. A threshold parameter is used to determine which logits are considered significant. By adjusting the threshold, one can control the trade-off between precision and recall, and therefore influence the overall F1 score.

3. Results and Discussion

The evaluation of our BERT-based model for Aspect-Based Sentiment Analysis (ABSA), specifically focused on extracting aspect terms, highlights the impact of threshold values used for identifying start and end indices on the model's performance metrics (see Table 2).
The results indicate that a threshold value around 1 provides the best balance between precision and recall, thereby optimizing the F1 score. Lower threshold values, such as -2 and -0.5, lead to an increased number of false positives, resulting in lower precision. On the other hand, higher thresholds, such as 1.5, 2, and 4, enhance precision but reduce recall, as the model becomes more conservative and may miss valid aspect terms, thereby decreasing the overall F1 score.

To further enhance the model's performance further, several approaches could be considered:

- Threshold Optimization: Conduct a more detailed analysis and implement heuristic algorithms to identify the optimal threshold based on the predicted start and end logits scores;
- Validation set: Create a validation set to monito and control model overfitting during training phase;
- Advanced Fine-Tuning: Experiment with different fine-tuning strategies, including learning rate adjustments, different optimization algorithms, and modifications to model architectures to better capture aspect terms.

4. References

[1] lixin4ever, "E2e-tbsa dataset," https://github.com/lixin4ever/E2E-TBSA/tree/master/data, 2020, accessed: 2024-07-31.

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

[3] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," 2019. [Online]. Available: https://arxiv.org/abs/1906.03820

[4] lixin4ever, "E2e-tbsa evaluation script," https://github.com/lixin4ever/E2E-TBSA/blob/master/evals.py, 2020, accessed: 2024-08-10.

1: *Details of the dataset.*

In this task, only the training and test sets were utilized, with no separate validation set created. The decision to omit the validation set was aimed at simplifying the workflow by directly assessing the model's final performance on the test set.

Table 1: *Laptop partition of SemEval2014 dataset statistics.*

	Train	**Test**
# sentences	3045	800
# sentences with no aspects	1587 (52%)	389 (49%)
# aspect terms	3408	1039
# T-POS	1367 (40%)	499 (48%)
# T-NEG	1300 (38%)	210 (20%)
# T-NEU	741 (22%)	330 (32%)
# aspects per sentence	1.12 ± 1.67	1.30 ± 1.92
Most Common Aspect Terms	battery: 97 screen: 82 use: 61 price: 57 Windows: 56	OS: 23 Windows: 17 price: 16 battery: 15 performance: 14

2: *Cross-Entropy loss.*
Where N is the batch size, L is the sequence length, $positions_{i,j}$ indicates whether position j in the sequence for example i is a true position, $log\_probs_{i,j}$ are the log probabilities for each position, $mask_i$ (optional) is used to exclude invalid positions and to handle division by zero cases.

$$loss = -\frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{L} positions_{i,j} * log\_probs_{i,j}}{\sum_{j=1}^{L} positions_{i,j} + mask_i}$$

3: *F1 equation: where P is precision* (4) *and R is recall* (5)

$$F1 = 2 * \frac{P * R}{P + R}$$

4: *Precision equation:*

$$Precision = \frac{T_P}{T_P + F_P}$$

5: *Recall equation:*

$$Recall = \frac{T_P}{T_P + F_N}$$

Table 2: *Results on train test. Analysis of the impact of different treshold values for the identification of Start and End indices on the fila result.*

Threshold	**F1**	**Precision**	**Recall**
-2	0.5618	0.6634	0.4872
-0.5	0.6119	0.7226	0.5307
1	**0.6407**	**0.7588**	**0.5545**
1.5	0.6369	0.7572	0.5497
2	0.6364	0.7580	0.5485
4	0.6345	0.7653	0.5420