NLU course projects - Assignment 3 - SA

Ettore Saggiorato (247178)

University of Trento

`ettore.saggiorato@studenti.unitn.it`

1. Introduction

This report discusses the implementation and evaluation of a model for aspect term extraction in the Aspect-Based Sentiment Analysis (ABSA) task. The model is fine-tuned using BERT [1] as part of the third assignment for the Natural Language Understanding course at the University of Trento. The `laptop14` partition from the SemEval-2014 Task 4 dataset [2] is used, which serves as a well-established benchmark for ABSA.

Performance is evaluated using standard metrics: precision, recall, and F1 score, computed using the official SemEval evaluation script [3, 4].

The goal is to fine-tune BERT to extract aspect terms, a task conceptually similar to slot filling, previously explored in the second course assignment. The similarity extends to the model architecture and training procedures, leveraging BERT's capabilities for token classification tasks.

2. Implementation Details

2.1. Dataset Preprocessing

The dataset consists of annotated sentences where each token is associated with a label. An example of a sentence annotation is provided below:

```
Boot time is super fast, around
anywhere from 35 seconds to
1 minute.  ####Boot=T-POS
time=T-POS is=O super=O fast=O
,=O around=O anywhere=O from=O
35=O seconds=O to=O 1=O minute=O
.=O
```

Each sentence is individually processed to map tokens and their corresponding tags into the BIESO tagging scheme (Begin, Inside, End, Single, Outside). This mapping is essential for compatibility with the official SemEval evaluation function, which requires this specific tagging format.

After preprocessing, the data is split into training, testing, and evaluation sets. Data loaders are configured to create batches dynamically, minimizing padding tokens. This optimization is crucial for efficient memory usage and helps reduce the presence of non-informative padding tokens, which could otherwise hinder the model's learning process.

The `bert-base-uncased` model and tokenizer are used, consistent with prior assignments. Tokens are processed using BERT's tokenizer to ensure compatibility with the model input format. In this approach, only the first subtoken of a word was assigned the relevant tag ID, while all subsequent subtokens were treated as padding (ID = 0). This effectively eliminated the evaluation distortion.

2.2. Fine-Tuning

The fine-tuning procedure follows the guidelines outlined in the BERT paper [1]. Training is conducted over four epochs. To ensure robust results, each experiment is repeated five times, and the outcomes are averaged for consistency.

To enhance training stability, the following regularization techniques are applied:

- **Gradient Clipping:** Gradient clipping is used to prevent exploding gradients, which can disrupt training, especially in deep networks like BERT.
- **Learning Rate Scheduler:** BERT's learning rate scheduler is employed to dynamically adjust the learning rate during training, ensuring smoother convergence and mitigating issues with sharp learning rate transitions.

3. Results

The model's performance is summarized in Table 1, using precision, recall, and F1 score as evaluation metrics. The key findings are as follows:

- Gradient Clipping: Gradient clipping significantly improves precision and F1 score, highlighting its role in stabilizing training and enhancing performance.
- Learning Rate Scheduler: The learning rate scheduler primarily improves recall, suggesting that dynamic adjustments to the learning rate allow the model to generalize better to harder-to-detect aspect terms.

Training the model for two additional epochs revealed improved performance, suggesting that the initial setting of four epochs was insufficient for optimal learning. These findings indicate that further training could yield even better results. However, additional epochs increase the risk of overfitting, which is particularly pronounced in this implementation. Therefore, applying stronger regularization techniques, such as dropout or weight decay, is essential to maintain generalization and prevent overfitting as training progresses.

4. References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] X. Li, "E2e-tbsa," https://github.com/lixin4ever/E2E-TBSA/tree/master/data.

[3] ——, "Evaluation script for semeval-2014 task 4," https://github.com/lixin4ever/E2E-TBSA/blob/master/evals.py.

[4] SemEval 2014 Organizers, "Semeval-2014 task 4: Aspect based sentiment analysis - data and tools," 2014. [Online]. Available: https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

Part	Model name	Epochs	Batch size	Gradient clip.	Train. sch.	F1 (%)	Precision (%)	Recall (%)
1.0	simple_16_2e-5	4	16	T	F	75.69	74.62	76.94
1.0	simple_16_3e-5	4	16	T	F	74.59	73.89	75.46
1.0	simple_16_5e-5	4	16	T	F	75.18	76.06	74.61
1.0	simple_32_2e-5	4	32	T	F	74.42	75.75	73.34
1.0	simple_32_3e-5	4	32	T	F	74.81	72.23	77.70
1.0	simple_32_5e-5	4	32	T	F	75.48	74.72	76.40
1.1	grad_16_2e-5	4	16	F	F	73.63	70.52	77.10
1.1	grad_16_3e-5	4	16	F	F	74.87	**77.56**	72.62
1.1	grad_16_5e-5	4	16	F	F	76.65	75.37	78.20
1.1	grad_32_2e-5	4	32	F	F	74.06	71.76	76.59
1.1	grad_32_3e-5	4	32	F	F	74.61	74.74	74.61
1.1	grad_32_5e-5	4	32	F	F	75.91	75.75	76.12
1.2	sch_grad_16_2e-5	4	16	T	T	76.50	74.66	**78.45**
1.2	sch_grad_16_3e-5	4	16	T	T	76.42	74.98	77.98
1.2	sch_grad_16_5e-5	4	16	T	T	76.42	76.42	78.33
1.2	sch_grad_32_2e-5	4	32	T	T	76.14	74.21	78.20
1.2	sch_grad_32_3e-5	4	32	T	T	**76.88**	75.63	78.20
1.2	sch_grad_32_5e-5	4	32	T	T	76.57	74.91	78.33
1.3	longer_grad_16_3e-5	6	16	F	T	75.60	**79.73**	76.56
1.3	longer_sch_grad_16_2e-5	6	16	T	T	**78.11**	77.17	79.09
1.3	***longer_sch_grad_16_5e-5***	6	16	T	T	*77.81*	*76.43*	***79.27***

Table 1: *Experiment Results.* ***Bold*** *values represent the most performant models for each part, italics represent the best averaged model overall.*