NLU Course Project - Aspect Based Sentiment Analysis

Christian Dalvit (249988)

University of Trento

christian.dalvit@studenti.unitn.it

1. Introduction

This project explores fine-tuning pre-trained BERT models for Aspect Based Sentiment Analysis, especially for target extraction. The effectiveness of these techniques is evaluated on the Laptop partition of SemEval2014 task 4 dataset using the precision, recall and F_1 score as a metric. The code of this project is made available on Github.

2. Implementation details

All models in this project are implemented in PyTorch [1]. The code implemented in the previous project about Natural Language Understanding was used as starting point for the implementation of this project. Mainly, the implementation for the PyTorch dataset and dataloader classes was adjusted. Additionally, the evaluation script from Tian et al. [2] was used and slightly adapted to exploit PyTorch tensor functionalities.

In this project, Aspect Based Sentiment Analysis and target extraction are formulated as slot filling task. Therefore, the Laptop partition of SemEval2014 task 4 is preprocessed into JSON files to fit the input format of the PyTorch dataset implementation from the previous project. For each sentence, a list of tokens with the corresponding labels is extracted. The possible labels are `O` (no target), `T-POS`, `T-NEU` or `T-NEG` (targets with positive, neutral or negative sentiment). The pretrained BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$ models [3] from Huggingface are used. A linear output layer is added after the BERT models for fine-tuning. An optional dropout layer is added before the final linear layer for regularization. Since Huggingface provides a convenient Python API, loading the different models was straightforward to implement. Both models process uncased input text.

As in the Natural Language Understanding project, the WordPiece [4] tokenization of BERT models can be problematic for slot filling tasks. In this project, the strategy of Chen et al. [5] is used and only the first subtoken of a word is used for Aspect Based Sentiment Analysis and target extraction. This is implemented by selecting the first subtoken of each word from the tokenized input text. The BERT models are fine-tuned on predicting targets and their corresponding sentiment. This collapsed annotation schema allows reporting scores for Aspect Based Sentiment Analysis and target extraction [6].

3. Results

All models were evaluated on the Laptop partition of SemEval2014 task 4 dataset, with the precision, recall and F_1 score used to measure slot filling performance. The evaluations were conducted on the Marzola cluster at the University of Trento. All models are fine-tuned with the AdamW [7] optimizer, cross-entropy loss, a training batch size of 16, gradient clipping and a dropout rate of 0.2 for 100 epochs, with early stopping patience of 5. Each setup is run 3 times.

Table 1: *Mean target extraction metrics*

Model	**Metrics**		
AdamW optimizer	**Precision**	**Recall**	**F_1**
BERT$_{\text{BASE}}$ 1×10^{-5}	89.87	82.96	86.26
BERT$_{\text{BASE}}$ 5×10^{-5}	90.98	81.81	85.96
BERT$_{\text{BASE}}$ 1×10^{-4}	91.94	79.66	85.35
BERT$_{\text{LARGE}}$ 1×10^{-5}	91.83	**83.16**	**87.24**
BERT$_{\text{LARGE}}$ 5×10^{-5}	91.48	82.87	86.86
BERT$_{\text{LARGE}}$ 1×10^{-4}	**92.61***	78.44*	84.93*

* This value was computed on only 1 run, because the traning procedure with the patience policy failed to converge to a good BERT$_{\text{LARGE}}$ model with a learning rate of 1×10^{-4}
Note: Bold values show the best score for each metric over all models.

As mentioned in Section 2 the models are fine-tuned on predicting the target with the corresponding sentiment. The metrics for target extraction are computed by only checking if the predicted label and the ground truth label start with `T`, hence ignoring the suffixes `-POS`, `-NEU` and `-NEG`. The evaluation script is adapted to provide metrics for target extraction and Aspect Based Sentiment Analysis. Table 1 reports mean precision, recall and F_1 score for the target extraction task. Generally, the BERT$_{\text{LARGE}}$ performs better compared to the BERT$_{\text{BASE}}$ model. This is probably a consequence of the higher number of parameters in BERT$_{\text{LARGE}}$.

Additionally, precision, recall and F_1 score for Aspect Based Sentiment Analysis are reported by Table 2. These metrics were computed by checking if the predicted labels exactly correspond with the ground truth labels, hence also considering the target sentiment. Predicting the full label is harder as target extraction, therefore all metrics in Table 2 are significantly lower compared to the target execution metrics. Similar to the target extraction task, BERT$_{\text{LARGE}}$ models perform better than BERT$_{\text{BASE}}$ models.

4. References

[1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[2] Y. Tian, W. Chen, B. Hu, Y. Song, and F. Xia, "End-to-end aspect-based sentiment analysis with Combinatory Categorial Grammar," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 597–13 609. [Online]. Available: https://aclanthology.org/2023.findings-acl.859

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert:

Table 2: *Mean sentiment analysis metrics*

Model	**Metrics**		
AdamW optimizer	**Precision**	**Recall**	$\mathbf{F_1}$
$\text{BERT}_{\text{BASE}}$ 1×10^{-5}	66.49	61.41	63.83
$\text{BERT}_{\text{BASE}}$ 5×10^{-5}	65.78	59.13	62.13
$\text{BERT}_{\text{BASE}}$ 1×10^{-4}	63.60	55.09	59.03
$\text{BERT}_{\text{LARGE}}$ 1×10^{-5}	69.63	**63.07**	**66.16**
$\text{BERT}_{\text{LARGE}}$ 5×10^{-5}	69.20	62.72	65.72
$\text{BERT}_{\text{LARGE}}$ 1×10^{-4}	**70.68***	59.86*	64.82*

* This value was computed on only 1 run, because the traning procedure with the patience policy failed to converge to a good $\text{BERT}_{\text{LARGE}}$ model with a learning rate of 1×10^{-4}
Note: Bold values show the best score for each metric over all models.

Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[4] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016. [Online]. Available: https://arxiv.org/abs/1609.08144

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

[6] 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

[7] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. [Online]. Available: https://arxiv.org/abs/1711.05101