Improving BERT Performance for Aspect-Based Sentiment Analysis

Akbar Karimi **Leonardo Rossi** **Andrea Prati**

University of Parma
{akbar.karimi, leonardo.rossi, andrea.prati}@unipr.it

Abstract

Aspect-Based Sentiment Analysis (ABSA) addresses the problem of extracting sentiments and their targets from opinionated data such as consumer product reviews. Analyzing the language used in a review is a difficult task that requires a deep understanding of the language. In recent years, deep language models, such as BERT, have shown great progress in this regard. In this work, we propose two simple modules called Parallel Aggregation and Hierarchical Aggregation to be utilized on top of BERT for two main ABSA tasks namely Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). With the proposed modules, we show that the intermediate layers of the BERT architecture can be utilized for the enhancement of the model performance[1].

1 Introduction

In an industry setting, it is extremely important to have a valid conception of how consumers perceive the products. Nowadays, they communicate their perception through their comments on the products, using mostly social networks. They might have positive opinions which can lead to the success of a business or negative ones possibly leading to its demise. Due to the abundance of these views in many areas, their analysis is a time-consuming and labor-intensive task which is why a variety of machine learning techniques such as Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Kiritchenko et al., 2014; Basari et al., 2013), Maximum Entropy (Jaynes, 1957; Nigam et al., 1999), Naive Bayes (Duda et al., 1973; Gamallo and Garcia, 2014; Dinu and Iuga, 2012), and Decision Trees (Quinlan, 1986; Wakade et al., 2012) have been proposed to perform opinion mining.

In recent years, Deep Learning (DL) techniques have been widely utilized due to the increase in computational power and the huge amount of freely available data on the Web (Zhang et al., 2015; Liu et al., 2015; Wang et al., 2016). One of the areas on which these techniques have had a great impact is Natural Language Processing (NLP) where modeling (i.e. understanding) the language plays a crucial role. BERT (Devlin et al., 2019) is a state-of-the-art model of this kind which has become widely utilized in many NLP tasks (Kantor et al., 2019; Davison et al., 2019) as well as in other fields (Peng et al., 2019; Alsentzer et al., 2019). It has been trained on a large corpus of Wikipedia documents and books in order to *learn* the language syntax and semantics from the context. The main component of its architecture is called the transformer (Vaswani et al., 2017) block consisting of attention heads. These heads have been designed to pay particular attention to parts of the input sentences that correspond to a particular given task (Vig and Belinkov, 2019). In this work, we utilize BERT for Aspect-Based Sentiment Analysis (ABSA) tasks.

Our main contribution is the proposal of two simple modules that can help improve the performance of the BERT model. In our models we opt for Conditional Random Fields (CRFs) for the sequence labeling task which yield better results. In addition, our experiments show that training BERT for more number of epochs does not cause the model to overfit. However, after a certain number of training epochs, the learning seems to stop.

2 Related Work

Recently, there has been a large body of work which utilizes the BERT model for various tasks in NLP in general such as text classification (Sun et al., 2019b), question answering (Yang et al.,

[1] https://github.com/IMPLabUniPr/BERT-for-ABSA

2019), summarization (Liu, 2019) and, in particular, ABSA tasks (Hoang et al., 2019).

Using Graph Convolutional Networks (GCNs), Zhao et al. (2020) take into account sentiment dependencies in a sequence. In other words, they show that when there are multiple aspects in a sequence, the sentiment of one of them can affect that of the other one. Making use of this information can increase the performance of the model. Some studies convert the Aspect Extraction (AE) task into a sentence-pair classification task. For instance, Sun et al. (2019a) construct auxiliary sentences using the aspect terms of a sequence. Then, utilizing both sequences, they fine-tune BERT on this specific task.

Word and sentence level representations of a model can also be enriched using domain-specific data. Xu et al. (2019) show this by post-training the BERT model, which they call BERT-PT, on additional restaurant and laptop data. In our experiments, we use their pre-trained model for the initialization of our models. Due to the particular architecture of the BERT model, extra modules can be attached on top of it. Li et al. (2019) add different layers such as an RNN and a CRF layer to perform ABSA in an end-to-end fashion. In our work, we use the same layer modules from the BERT architecture and employ the hidden layers for prediction as well.

3 Aspect-Based Sentiment Analysis Tasks

Two of the main tasks in ABSA are Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). While the latter deals with the semantics of a sentence as a whole, the former is concerned with finding which word that sentiment refers to. We briefly describe them in this section.

3.1 Aspect Extraction

In AE, the goal is to extract a specific aspect of a product towards which some type of sentiment is expressed in a review. For instance, in the sentence, "*The laptop has a good battery.*", the word *battery* is the aspect which is extracted. Sometimes, the aspect words can be multiple in which case all of them need to be labeled accordingly. This task can be seen as a sequence labeling task, where the words are assigned a label from the set of three letters namely $\{B, I, O\}$. Each word in the sequence can be the beginning word of aspect terms (B), among the aspect terms (I), or not an aspect term (O). The classification of each word into one of these three classes, is accomplished using a fully connected layer on top of the BERT architecture and applying the Softmax function.

3.2 Aspect Sentiment Classification

In this task, the goal is to extract the sentiment expressed in a review by the consumer. Given a sequence, one of the three classes of *Positive*, *Negative*, and *Neutral* is extracted as the class of that sequence. The representation for this element is embodied in the architecture of the BERT model. For each sequence as input, there are two extra tokens that are used by the BERT model:

$$[CLS], w_1, w_2, ..., w_n, [SEP]$$

where w_i are the sequence words and $[CLS]$ and $[SEP]$ tokens are concatenated to the sentence in the input stage. While the $[CLS]$ token is there to store the sentiment representation of the sentence, the $[SEP]$ token is used to separate input sequences in case there are more than one (e.g. in a question answering task). In the final layer of the architecture, a Softmax function is applied to the $[CLS]$ embedding and the class probability is computed.

4 Proposed Model

Deep models can capture deeper knowledge of the language as they grow. As shown by Jawahar et al. (2019), the initial to middle layers of BERT can extract syntactic information, whereas the language semantics are represented in higher layers. Since extracting the sentence sentiment is semantically demanding, we expect to see this in higher layers of the network. This is the intuition behind our models where we exploit the final layers of the BERT model.

The two models that we introduce here are similar in principle, but slightly differ in implementation. Also, for the two tasks, the losses are computed differently. While for the ASC task we utilize cross-entropy loss, for the AE task, we make use of CRFs. The reason for this choice is that the AE task can be treated as sequence labeling. Therefore, taking into account the previous labels in the sequence is of high importance, which is exactly what the CRF layer does.

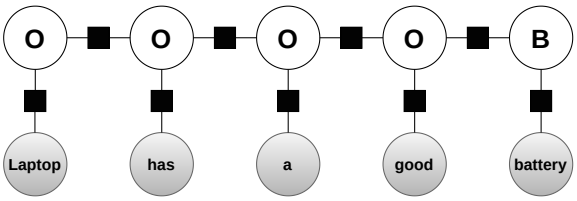

Figure 1: An example of representing a sentence with its word labels using CRFs.

4.1 Conditional Random Fields

CRFs (Lafferty et al., 2001) are a type of graphical models and have been used both in computer vision (e.g. for pixel-level labeling (Zheng et al., 2015)) and in NLP for sequence labeling.

Since AE can be considered a sequence labeling task, we opt for using a CRF layer in the last part of our models. The justification for the use of a CRF module for AE is that doing so helps the network to take into account the joint distribution of the labels. This can be significant since the labels of sequence words are dependent on the words that appear before them. For instance, as is seen in Figure 1, the occurrence of the adjective *good* can give the model a clue that the next word is probably not another adjective. The equation with which the joint probability of the labels is computed is as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\} \quad (1)$$

In Formula 1, $\mathbf{x}$ is the observed sequence, $\mathbf{y}$ is the sequence of labels, and k and t are the indices for feature functions and time steps in the sequence, respectively. The relations between sequence words are represented by using feature functions $\{f_k\}$. These relations can be strong or weak, or non-existent at all. They are controlled by their weights $\{\theta_k\}$ which are computed during the training phase. Finally, $Z(\mathbf{x})$ is a normalization factor.

4.2 Parallel Aggregation

Rossi et al. (2020) showed that the hidden layers of deep models can be exploited more to extract region specific information. Inspired by their work, we propose a model called P-SUM applying BERT layer modules on each one of the best performing BERT layers. Figure 2 shows the details of this model. We exploit the last four layers of the BERT model by adding one more BERT layer plus a fully connected layer and calculating the loss of that branch on the input data, using a Softmax function

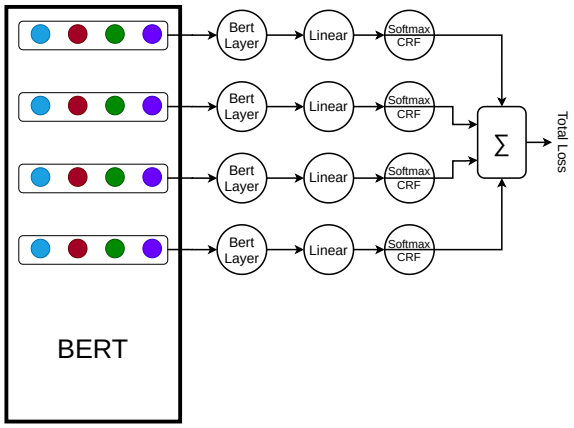

Figure 2: Parallel aggregation (P-SUM)

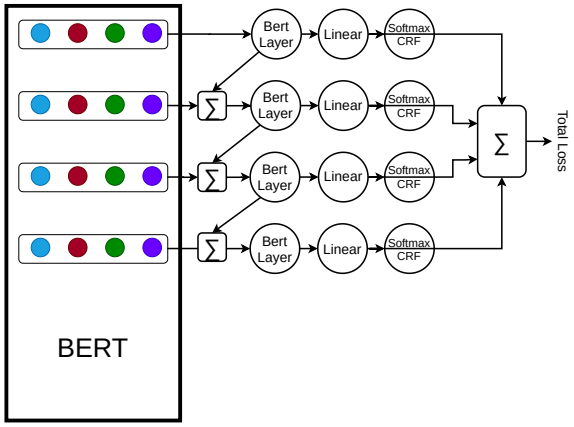

Figure 3: Hierarchical aggregation (H-SUM)

and a conditional random fields layer. The reason is that all deeper layers contain most of the related information regarding the task. Therefore, extracting this information from each one of them and combining them can produce richer representations of the semantics. In order to calculate the total loss, the loss values of all branches are summed up which is indicated with Σ notation in the diagram. This is done so, in order to take all the losses into account when optimizing the parameters. However, to compute the network's output $logits$, we average over the output $logits$ of the four branches.

4.3 Hierarchical Aggregation

Our hierarchical aggregation (H-SUM) model is inspired by the use of Feature Pyramid Networks (FPNs) (Lin et al., 2017). The goal is to extract more semantics from the hidden layers of the BERT model. The architecture of the H-SUM model can be seen in Figure 3. Here, after applying a BERT layer on each one of the hidden layers, the output is aggregated (element-wise) with the previous

	Train		Test	
Dataset	S	A	S	A
LPT14	3045	2358	800	654
RST16	2000	1743	676	622

Table 1: Laptop (LPT14) and restaurant (RST16) datasets from SemEval 2014 and 2016, respectively, for AE. S: Number of sentences; A: Number of aspects.

	Train				Test			
Dataset	S	Pos	Neg	Neu	S	Pos	Neg	Neu
LPT14	2313	987	866	460	638	341	128	169
RST14	3102	2164	805	633	1120	728	196	196

Table 2: Laptop (LPT14) and restaurant (RST14) datasets from SemEval 2014 for ASC. S: Number of all sentences; Pos, Neg, Neu: Number of positive, negative, and neutral sentiments, respectively.

layer. At the same time, similar to the P-SUM, each branch produces a loss value which contributes to the total loss equally since the total loss is the summation of all of them.

5 Experiments

In order to carry out our experiments, we use the same codebase as Xu et al. (2019). We ran the experiments on a GPU (GeForce RTX 2070) with 8 GB of memory using batches of 16 for both our models and the BERT-PT model as the baseline. For training, Adam optimizer was used and the learning rate was set to $3e-5$. From the distributed training data, we used 150 examples as the validation. To evaluate the models, the official scripts were used for the AE tasks and the script from the same codebase was used for the ASC task. Results are reported in F1 for AE and in Accuracy and MF1 for ASC. While F1 score is the harmonic mean of precision and recall, MF1 score is the average of F1 score for each class.

5.1 Datasets

In our experiments, we utilized laptop and restaurant datasets from SemEval 2014 (Pontiki et al., 2014) Subtask 2 and 2016 (Pontiki et al., 2016) Subtask 1. The collections consist of user reviews which have been annotated manually. Tables 1 and 2 show the statistics of these datasets. In choosing the datasets, we opted for the ones utilized in previous works (Karimi et al., 2020; Xu et al., 2019) so that we can draw a reliable comparison between the performance of our models and those ones.

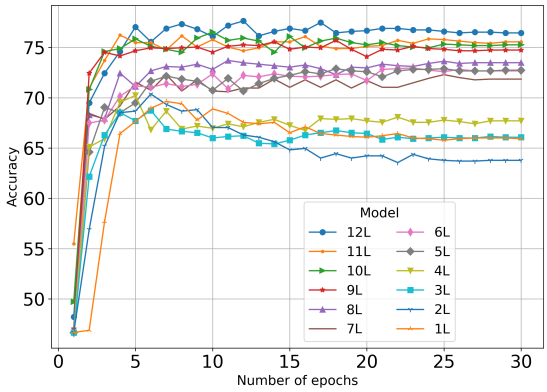

Figure 4: Performance of BERT layers initialized by BERT-PT weights for ASC on RST14 validation data. Each model is the BERT model using the specified number of layers. 1L means using the first layers, 2L means using the first 2 layers, etc. Accuracy values are percentages.

5.2 Performance of BERT Layers

Depending on the depth of the network, it can perform differently. Therefore, we carried out experiments to find out how each layer of the BERT model performs. The results are shown in Figure 4. As can be seen, better performance is achieved in the deeper layers, especially the last four. Therefore, our modules operate on these four layers to achieve an improved model.

5.3 Increasing Training Epochs

More training can lead to a better performance of the network. However, one risks the peril of overfitting especially when the number of training examples are not considered to be large compared to the number of parameters contained in the model. However, in the case of BERT, as was also observed by Li et al. (2019), it seems that with more training the model does not overfit although the number of the training data points is relatively small. The reason behind this could be the fact that we are using an already pre-trained model which has seen an enormous amount of data (Wikipedia and Books Corpus). Therefore, we can expect that by performing more training, the model will still be able to generalize.

The same observation can be made by looking at the validation losses in Figure 5. In case of an overfit, we would expect the losses to go up and the performance to go down. However, we see that with the increase in loss after the second epoch, the

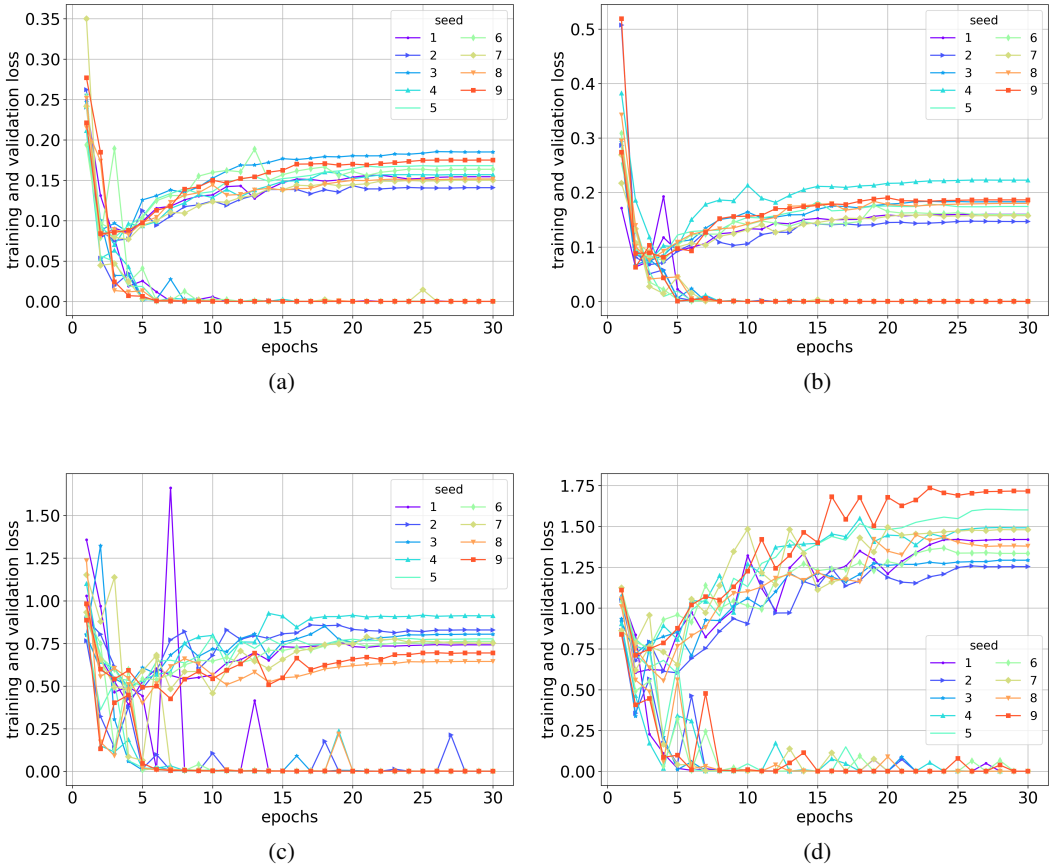

Figure 5: Training and validation losses of the 12-layer BERT model initialized with BERT-PT weights for AE (laptop (a) and restaurant (b)) and ASC (laptop (c) and restaurant (d)). In each figure, the upper lines are validation losses and the bottom lines are training losses, each line corresponding to a seed number.

performance still improves for a couple of epochs and then fluctuates in the subsequent ones (Figure 4). This suggests that with more training, the network weights continue to change until they remain almost stable in later epochs, indicating that there is no more learning. From Figure 4, we see that with 4 or 5 training epochs we get near the maximum performance. Although some later epochs such as 12 yield better results for the 12-layer version, it can be considered negligible.

6 Results

Our experimental results show that with the increase of the training epochs the BERT model also improves. These results can be seen in Table 3. To compare our proposed models with Xu et al. (2019), we perform the same model selection for both of them. Unlike Xu et al. (2019) and Karimi et al. (2020) who select their best models based on the lowest validation loss, we choose the models trained with four epochs after observing that accuracy goes up on the validation sets (Figure 4). Therefore, in Table 3, we report the original BERT-PT scores as well as the ones for our model selection. From Table 3, it can also be seen that the proposed models outperform the newly selected BERT-PT model in both datasets and tasks with improvements in MF1 score as high as **+1.78** and **+2** for ASC on laptop and restaurant, respectively.

It is also worth noting that, in terms of accuracy, the H-SUM module performs better than the P-SUM module in most cases. This could be attributed to the hierarchical structure of the module and the fact that each branch of this module benefits from the information processed in the preceding branch.

7 Conclusion

We proposed two simple modules utilizing the hidden layers of the BERT language model to produce

	AE		ASC			
	LPT14	RST16	LPT14		RST14	
Models	F1	F1	Acc	MF1	Acc	MF1
BERT	79.28	74.10	75.29	71.91	81.54	71.94
DE-CNN (Xu et al., 2018)	81.59	74.37	-	-	-	-
BERT-PT (Xu et al., 2019)	84.26	77.97	78.07	75.08	84.95	76.96
BAT (Karimi et al., 2020)	85.57	81.50	79.35	76.50	86.03	79.24
BERT-PT*	85.57	81.57	78.21	75.03	85.43	77.68
P-SUM	**85.94**	**81.99**	**79.55**	**76.81**	**86.30**	**79.68**
H-SUM	**86.09**	**82.34**	**79.40**	**76.52**	**86.37**	**79.67**

Table 3: Comparison of the results for Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). BERT-PT* is the original BERT-PT model using our model selection. The boldfaced numbers show the outperforming models using the same settings. Each score in the table is the average of 9 runs. Results for the cited papers are reported from the corresponding paper. The other models are run for 4 epochs. LPT: Laptop, RST: Restaurant, Acc: Accuracy , MF1: Macro-F1. Values are percentages.

deeper semantic representations of input sequences. The layers are once aggregated in a parallel fashion and once hierarchically. For each branch of the architecture built on top of the selected hidden layers, we compute the loss separately. These losses are then aggregated to produce the final loss of the model. We address aspect extraction using conditional random fields which helps to take into account the joint distribution of the sequence labels to achieve more accurate predictions. Our experiments show that the proposed approaches outperform the post-trained vanilla BERT model.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Abd Samad Hasan Basari, Burairah Hussin, I Gede Pramudya Ananta, and Junta Zeniarja. 2013. Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, 53:453–462.

Corinna Cortes and Vladimir Vapnik. 1995. Support vector machine. *Machine learning*, 20(3):273–297.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Liviu P Dinu and Iulia Iuga. 2012. The naive bayes classifier in opinion mining: in search of the best feature set. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 556–567. Springer.

Richard O Duda, Peter E Hart, et al. 1973. *Pattern classification and scene analysis*, volume 3. Wiley New York.

Pablo Gamallo and Marcos Garcia. 2014. Citius: A naive-bayes strategy for sentiment analysis on english tweets. In *Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014)*, pages 171–175.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review*, 106(4):620.

Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. Learning to combine grammatical error corrections. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148.

Akbar Karimi, Leonardo Rossi, Andrea Prati, and Katharina Full. 2020. Adversarial training for aspect-based sentiment analysis with BERT. *arXiv preprint arXiv:2001.11316*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1433–1443.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Kamal Nigam, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 27–35.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

Leonardo Rossi, Akbar Karimi, and Andrea Prati. 2020. A novel region of interest extraction layer for instance segmentation. *arXiv preprint arXiv:2004.13665*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of NAACL-HLT*, pages 380–385.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76.

Shruti Wakade, Chandra Shekar, Kathy J Liszka, and Chien-Chung Chan. 2012. Text mining for sentiment analysis of twitter data. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer ....

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data augmentation for BERT fine-tuning in open-domain question answering. *arXiv preprint arXiv:1904.06652*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 649–657.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowledge-Based Systems*, 193:105443.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537.