Utilizing BERT pretrained models with various fine-tune methods in subjectivity tasks

Hairong Huo † and Mizuho Iwaihara ‡

Graduate School of Information, Production and Systems, Waseda University 2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0135 Japan

E-mail: † hairong.huo@ruri.waseda.jp, ‡ iwaihara@waseda.jp

Abstract As an essentially antecedent task of sentiment analysis, subjectivity detection refers to classifying sentences to be subjective ones containing opinions, or objective and neutral ones without bias. In the situations where impartial language is required, such as Wikipedia, subjectivity detection could play an important part. Recently, pretrained language models have proven to be effective in learning representations, profoundly boosting the performance among several NLP tasks. As a state-of-art pretrained model, BERT is trained on large unlabeled data with masked word prediction and next sentence prediction tasks. In this paper, we mainly explore utilizing BERT pretrained models with several combinations of fine-tuning methods, holding the intention to enhance performance in subjectivity detection task. Our experimental results reveal that optimum combinations of fine-tune and multi-task learning surplus the state-of-the-art on subjectivity detection and related tasks.

Keyword Subjectivity Task, Fine-tuning, Pretrained Model, BERT

1. Introduction

In natural language processing research, a wide variety of methods have been attempted to interpret information implied in written texts, since knowing the ideas and minds hidden behind the texts is essential to profoundly understand our life in many aspects. Among all the research directions, *sentiment analysis*, also known as opinion mining, is the study field on estimating people's opinions, sentiments, feelings, as well as attitudes towards objects, news, issues, markets, etc [9][29]. Recently, with the dramatic increase of opinionated statements, growing research effort has been paid on sentiment analysis as well as its subtask, *subjectivity detection* [14]. As an essentially antecedent task of sentiment analysis, subjectivity detection task seeks classifying a sentence into objective and neutral ones without any bias, or instead, subjective and biased ones [15].

As examples of subjective language introducing bias and objective one, consider the following statements:

• *Scientologists hold the belief that living cells have a memory. This is based on an* ***erroneous*** *interpretation of the work of Crick and Watson in 1955.* ***(opinion, not a fact)***

• *Scientologists hold the belief that living cells have a memory. This is based on an interpretation of the work of Crick and Watson in 1955.* ***(fact, not an opinion)***

In the above instances, the word ***erroneous*** introduces bias, causing the statement being partial.

Generally, one has to classify a sentence as subjective or objective, then the resulting subjective sentence is classified as positive or negative [29]. Also, in collaborative environments where people around the world share information upon, such as Wikipedia, fair-and-square language are desired [6][8]. Moreover, scenarios like news report will require content to be impartial and deliver objective information to readers. All these reasons make subjectivity detection of vital essence in NLP research area.

Although many researches utilize deep learning models to achieve the state-of-art on many NLP tasks as well as subjectivity detection task here, these models require large amounts of datasets, training time as well as computational resources to train from scratch. Alternatively, plentiful researches have proven that pretraining language models based on large corpus and fine tuning them on task specific datasets can be beneficial for various NLP tasks including subjectivity detection task [5][7]. The concept and methodology have been widely used in the computer vision (CV) area. By merely fine tuning the pre-trained model based on a large dataset such as ImageNet can generate great results on a specific task without training everything from scratch. Inspired by the benefits of pretraining, various carefully designed language models have recently emerged, such as OpenAI GPT [20], UMLFit [7] and BERT [5]. Built on multi-layer bidirectional Transformer [25] blocks, BERT is trained on huge datasets based on two tasks: masked word prediction and next sentence prediction. As the first

fine-tuning based representation model, BERT has showed its effectiveness in several NLP tasks. However, its potential has not been thoroughly explored, which leaves us space to search further.

To this end, the contributions of our paper are as follows:

•We discuss utilizing the BERT pretrained language model to fine-tune toward the subjectivity detection task.

•We then further explore several methods to fine-tune BERT for subjectivity detection and related tasks. Also, influence of different combinations of fine-tuning methods for performance is investigated.

The rest of this paper is organized as follows: Section 2 covers related work. Section 3 describes methodologies. Section 4 shows experimental results, and Section 5 is a conclusion.

2. Related work

2.1 Subjectivity detection

Although compared to sentiment analysis, researches conducted for subjectivity task are relatively less. There exist outstanding works regarding the subjectivity task. Chenghua Lin et al. [11] present a hierarchical Bayesian model based on latent Dirichlet allocation (LDA) for subjectivity detection. Instead of designing models based on a pre-labelled dataset or linguistic pattern extraction, they regard the subjectivity task as weakly-supervised generative model learning. Moreover, as the largest collaborative online encyclopedia characterized by free editorial content around the world, there are substantial works [1][6][8][17] conducted on Wikipedia for distinguishing biased statements from impartial language. Desislava et al. [1] propose a multilingual method for detection of biased statements in Wikipedia and creates corpora in Bulgarian, French and English. They utilize a multinomial logistic regression algorithm on top of pretrained word embeddings. Christoph et al. [6] propose a feature-based supervised classification method to detect biased and subjective language in Wikipedia. They achieved detection accuracy of 74% on a dataset consisting of biased and unbiased statements. However, utilizing manually constructed features can be incomprehensive, and time and resource consuming. Christoph et al. [8] present a neural based model with hierarchical attention mechanism to solve the problem. In their work, they first crawl Wikipedia revision history that have a “POV” flag, that is “point of view,” suggesting certain statements containing opinions and subjective ideas towards entities. Additionally, to improve the quality of the original dataset, they use crowdsourcing to filter statements that do not contain bias and subjective information. They finally release the largest corpus of statements annotated for biased language and are able to distinguish biased statements with a precision of 0.917.

2.2 Pretrained language model

Although deep neural models can be impressive in the related researches, it would take too much computational resource and time to converge, as well as demanding large labeled datasets to train from scratch [24]. Thus, utilizing pretrained representations and fine-tuning methods can alleviate the problem. Howard et al. [7] propose a universal language model fine-tuning (UMLFit), pretrained on Wikitext-103 consisting of 28,595 Wikipedia articles to capture semantic and syntactic information. Their method significantly surpasses the state-of-art models on six text classification tasks. Meanwhile Devlin et al. [5] release BERT, the first fine-tuning based representation model that achieves the state-of-the art results on various NLP tasks, making a huge breakthrough in related research areas. Trained on a large cross-domain corpus, BERT is designed for two pretrained tasks: masked language model task and next sentence prediction task. Different from UMLFit, BERT is not limited to the simple combination of two unidirectional language models. Instead, BERT utilizes masked language model to predict words which are masked at random to capture bidirectional and contextual information.

Indeed, BERT is a state-of-art model outperforming in a variety of NLP tasks, demonstrating its effectiveness and potential. In this paper, we aim to explore the fine-tuning methods of BERT in subjectivity detection task, with intention to explore optimum fine-tuning strategies.

3. Methodologies

One of the most remarkable features about BERT is that merely utilizing the released BERT model by Google AI and fine-tuning it can generate relatively good results, especially on small datasets, like the case in subjectivity detection task.

3.1 How to fine-tune BERT for subjectivity task?

A BERT-base model consists of a large encoder built with 12 transformer blocks and 12 self-attention heads, with hidden size of 768. The input of BERT Model is a sequence with length no longer than 512 tokens while the

output of BERT is the representation of the whole sequence. In the meanwhile, there are two special tokens in BERT: [CLS], which contains the classification embedding information, while token [SEP] is utilized for separating segments of input [5]. Our goal is to separate subjective statements with bias from objective and unbiased ones. For this kind of single sentence classification problem, we can simply plug task-specific inputs into the BERT architecture, and after multi-layer transformer blocks, the final hidden state h of the first token [CLS] in the last layer can be viewed as the ultimate representation of the whole sequence. Then, whether a simple classifier like softmax or other more complicated methods like Long Short Term Memory Network (LSTM) can be added upon the top of BERT to do a classification task.

3.2 Layer-wise discriminative fine-tuning

In addition to applying a simple classifier like softmax or other more complex ones such as LSTM, we further explore several fine-tuning strategies to help improve the performance. The first method is *layer-wise discriminative fine-tuning* [7]. A BERT model contains a deep encoder consisting of 12 transformer blocks, in other words, the BERT model has 12 layers and each of them is responsible to capture information with different extent. As a matter of course, these layers should be fine-tuned with different extent consistently. To this end, layer-wise discriminative learning rate for each layer is necessary. Instead of allocating all layers with a same learning rate like typical regular stochastic gradient descent (SGD), following Howard and Ruder [7], we choose to give each layer a different learning rate. In regular stochastic gradient descent, the parameters are updated by the following equation:

$$\theta_t = \theta_{t-1} - \eta.\nabla_\theta J(\theta), \quad (1)$$

where η is the learning rate, and $\nabla_\theta J(\theta)$ is the gradient related to the model's objective function. As for layer-wise discriminative learning rate, we replace η with multiple learning rates $\{\eta^1, \dots, \eta^L\}$, where η^l denotes the learning rate of l-th layer and L is the total number of the layers. Similarly, we can obtain the parameters $\{\theta^1, \dots, \theta^L\}$, where θ^l consists of the parameters of the l-th layer. By using layer-wise discriminative learning rate, the update of the parameters can be showed as follows:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \nabla_{\theta^t} J(\theta) \quad (2)$$

During the experiment part, we set the initial learning rate as 2e-5, and utilize $\eta^{l-1} = \eta^l / 1.1$ as the learning rate for lower layers. Thus, the lower layers tend to have a lower learning rate than higher layers. Intuitively, the lower layers of BERT may contain more general information while higher layer contains more specific information.

3.3 One cycle policy

Learning rate is an essential hyperparameter in the neural network, but how to choose an appropriate learning rate can be a subtle and tricky problem, which has perplexing the researchers for a long time. The small learning rate might make the model to converge slowly, leading to a long training time, while a large one may also contribute to diverging.

Leslie N. Smith [23] proposes the strategy *one cycle policy*. In simple terms, one-cycle-policy uses a periodic learning rate. The motivation behind one cycle policy is that during the middle of learning when learning rate is higher, the learning rate works as regularization method to some extent and keep model away from over-fitting. This helps model to avoid steep areas of loss and land better as well as flatter minima. It is a modification of the cyclical learning rate policy [22]. But one cycle policy allows a large initial learning rate (e.g: $LR_{MAX} = 10^{-3}$). This seems to provide greater accuracy.

The schedule of implementing one cycle policy learning rate strategy is described as below:

(1) Initial learning rate. Following the study of Leslie N. Smith [22], we first choose the maximum learning rate according to the *LR range test*. The idea here is that we need to use a learning rate (lr_{max}) in an order of magnitude lower than the point where the loss of the model starts to diverge. That is, if the learning rate is below 0.01, the loss of the model starts to diverge, then 0.01 should be the initial learning rate lr_{max}. After choosing the appropriate initial learning rate lr_{max}, we then set the minimum learning rate lr_{min} equal to 1/10 of maximum learning rate lr_{max}.

$$lr_{min} = \frac{1}{10} * lr_{max}. \quad (3)$$

(2) Cyclical momentum. After setting the initial learning rate lr_{max}, we then gradually increase the learning rate from lr_{min} to lr_{max} by utilizing the *cyclical momentum*. According to Leslie [23], decreasing momentum while increasing learning rate leads to better result. In the experiment part, we pick two values for maximum and minimum momentum: 0.9 and 0.8. As we increase the learning rate from lr_{min} to lr_{max}, the momentum is decreased from mom_{max} to mom_{min} (warm-up step). Then go back to the higher momentum as the learning rate

goes down (cool-down step).

(3) Annihilation phase. After the warm-up and cool-down phase, the third phase is annihilation. As the last part of training, we decrease the learning rate up to a value equal to 1/100 of minimum learning rate and keep the momentum steady at mom_{max}.

$$\begin{cases} lr_{annihilation} = \frac{1}{100} * lr_{min} \\ mom_{annihilation} = mom_{max} \end{cases} \quad (4)$$

The following image shows the one cycle policy learning rate strategy:

Figure 1: 1 cycle policy learning rate strategy

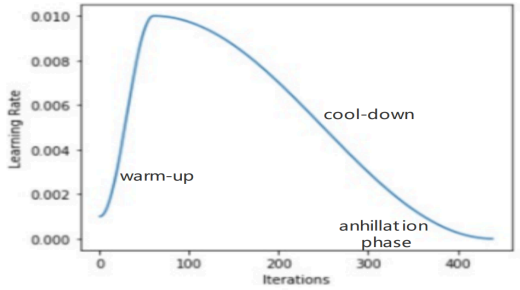
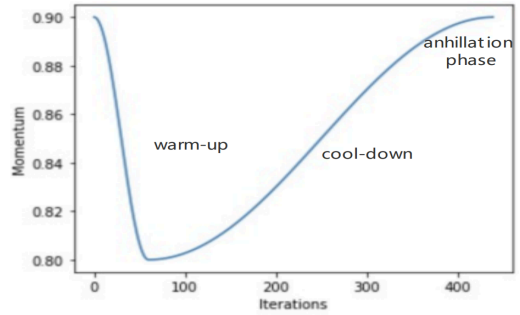


Figure 2: 1 cycle policy momentum strategy



3.4 Gradual unfreezing

In transfer learning, there is a common problem of catastrophic forgetting, which refers to the phenomenon such that pretrained knowledge is lessened during the process of learning new knowledge. To overcome this problem, we adopt the strategy called *gradual unfreezing* [7]. For gradual unfreezing, rather than fine-tuning all layers at one time, which is likely to lead to catastrophic forgetting, we first unfreeze the last layer of BERT and fine tune for one epoch, while the remained layers are frozen. Then, we unfreeze the next frozen layer and fine tune all the unfrozen layers. The rest can be done in the same manner.

3.5 Multi-task learning

3.5.1 Motivations

In the research field of transfer learning, there has always been a prevalent interest in multi-task learning, since by utilizing multi-task learning instead of training single task separately, the performance has been improved in vast domains from computer vision to natural language processing [3][28]. With the purpose of improving the performance as well as enhancing the learning efficiency, multi-task learning refers to learning several tasks jointly, so that the knowledge learned in a single task can benefit other tasks. Generally, as soon as optimizing more than one loss function is done, it is effectively equivalent to do multi-task learning in contrast to single-task learning [19].

The motivations behind multi-task learning can be divided into the following aspects.

•**Data augmentation:** Traditional supervised neural networks require large amounts of labelled datasets to train from scratch, especially for the deep and complex networks. However, the chances are that such large scale datasets cannot always be available. By incorporating several tasks as well as datasets, the sample size used for training can be enlarged, which can be helpful for the low-resource task.

•**Regularization:** Multi-task learning can provide the regularization effect to some extent by introducing an inductive bias. Learning single task tends to bear the risk of overfitting, while learning several tasks simultaneously enables levitating the problem, leading to a better universal representation through all the tasks.

In the meanwhile, pretrained language models have proven to be effective in learning universal representations by leveraging plenty of unlabeled data, such as BERT [5]. To apply released BERT pretrained language model to specific task, traditionally, we often fine-tune BERT for each task separately with task-specific layer and training datasets. There has been several studies [12][24] arguing that multi-task learning and pretrained language model are complementary technologies and thus, the two of them can be combined together to boost the performance. To this end, we utilize multi-task learning strategy with shared BERT representation layers, with the intention to enhance the performance of our main task: subjectivity detection.

3.5.2 Tasks

We list several NLP tasks as follow as our auxiliary tasks, with the intention to generate a better universal representation and improve the performance.

• **Text similarity task:** For text similarity task, its goal is to determine the similarity of two given texts. As a regression task, the output should be a real-value score, indicating the semantic similarity of two pieces of texts. Given a pair of texts (x_1, x_2), to apply BERT to the text

similarity task, we can take the final hidden state x of token [CLS] in BERT structure, which can be viewed as the representation of the whole sequence pair (x_1, x_2), the similarity score can be computed as follow:

$$SIM_{(x_1,x_2)} = W_{SIM}^T \cdot x \quad (5)$$

where W_{SIM}^T is the task specific parameter matrix. Since the text similarity task is a regression task, we utilize mean squared error as the loss function:

$$LOSS = \frac{1}{n}\sum_{i=1}^{n}(y - SIM_{(x_1,x_2)})^2 \quad (6)$$

•Pair-wise text classification task: Pair-wise text classification task refers to predicting the relationship between two texts based on a set of predefined labels. When applying BERT to this task, given a pair of texts (x_1, x_2) and x denotes the final hidden state of [CLS] token, the probability that x is labeled as class c (i.e., entailment) is predicted by a softmax classifier:

$$P(c|x) = softmax(W_{pairwise}^T \cdot x) \quad (7)$$

where $W_{pairwise}^T$ is the task specific parameter matrix. We utilize cross entropy as the loss function:

$$LOSS = -\sum_i y_i \cdot \log(P(c|x)) \quad (8)$$

•Pair-wise relevance ranking task: Given a query and a list of candidate answers, the goal of pair-wise relevance ranking task is to rank all the candidates in the order of relevance to the query. When applying BERT to this task, suppose that x denotes the final hidden state of [CLS] token, and the input is a pair of query Q and candidates collection A, the relevance score can be computed as following equation:

$$Relevance(Q, A) = g(W_{relevance}^T \cdot x) \quad (9)$$

where $W_{relevance}^T$ is the task specific parameter matrix. For a given query, the candidates can be ranked on the basis of relevance score. Following the study of Liu et al. [12], we utilize the pairwise learning-to-rank paradigm [2]. Given a query Q, we can obtain a list of candidate answers A which contains a positive example A^+ that includes the correct answer, and $|A| - 1$ negative examples. We then minimize the negative log likelihood of the positive example given queries across the training data:

$$LOSS = -\sum_{(Q,A^+)} P_r(A^+|Q) \quad (10)$$

$$P_r(A^+|Q) = \frac{\exp(Relevance(Q,A^+))}{\sum_{A' \in A}\exp(Relevance(Q,A'))} \quad (11)$$

•Single sentence classification task: This task refers to classifying a single sentence into a pre-defined label. As an important research branch of single sentence classification, subjectivity detection task is our main task. To apply BERT to this task, similarly, we take the final hidden state x of [CLS] token as the representation of the whole sequence. The probability that x is labeled as class c (i.e., subjective) is predicted by a softmax classifier:

$$P(c|x) = softmax(W_{classification}^T \cdot x) \quad (12)$$

where $W_{classification}^T$ is the task specific parameter matrix. For the classification problem, we utilize cross entropy as our loss function:

$$LOSS = -\sum_i y_i \cdot \log(P(c|x)) \quad (13)$$

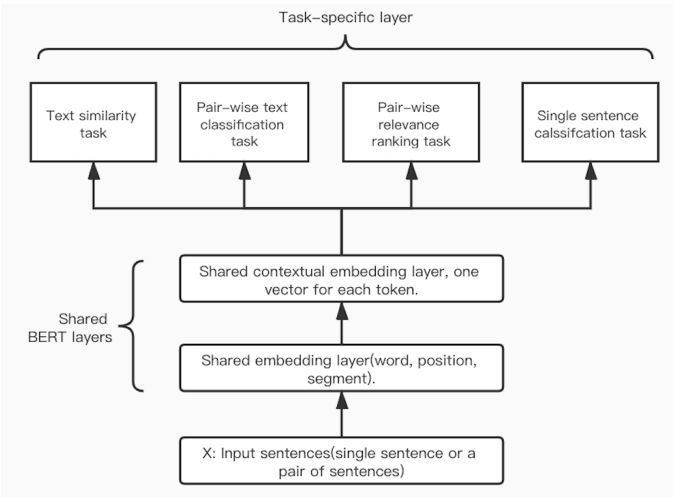
3.5.3 Multi-task learning with shared BERT layer

•Utilizing BERT-base model as pretrained representation: The architecture of combining multi-task learning with BERT as shared representation layer is showed in Figure 3, where the lower layers are shared among four tasks, while the top layers are task-specific layers, corresponding to four tasks. The input of the shared BERT layer can be a single sentence or a pair of sentences, then, the sentences will be first represented as a set of embedding vectors, consisting of the word, position and segment embedding. Then, BERT is responsible for capturing the contextual information, and the contextual embeddings will be generated.

•Multi-task learning upon BERT: The above shared BERT layers are task specific layers corresponding to different tasks. Our main task is the single sentence classification. By incorporating BERT with multi-task learning, our goal is to further enhance the performance for subjectivity detection, since compared to other tasks, subjectivity task can be viewed as a task that lacks sufficient labelled samples, where multi-task learning strategy might play a role.

•Further fine-tuning with task-specific dataset: In order to improve the performance, we further fine-tune the learned model with task-specific dataset to generate a final result.

Figure 3: Multi-task learning with shared BERT layer



4. Experiments

4.1 Datasets

STS-B dataset [4]: The Semantic Textual Similarity Benchmark (STS-B) dataset is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Given a pair of texts (x_1, x_2), the goal is to compute the similarity between x_1 and x_2, returning a similarity score ranging from 0 to 5. The dataset consists of 8,628 sentences pairs.

SNLI dataset [21]: The Stanford Natural Language Inference (SNLI) dataset, consisting of 570,152 sentence pairs, is a collection written by human, and is manually labeled with pre-defined three labels: entailment, neutral and contradiction. Given a pair of texts (x_1, x_2), the task if to predict the relationship between them and output the label among entailment, neutral and contradiction.

QNLI dataset [18]: The Stanford Question Answering (QNLI) dataset is a question-answering dataset consisting of 115,669 question-paragraph pairs. While the question is manually written by an annotator, the paragraph is drawn from Wikipedia containing the answer to the question. Although QNLI is originally defined as a binary classification task to predict whether the paragraph contains the answer to corresponding question or not, following Liu et al. [12], in this study, we formulate it into pair-wise relevance ranking task to improve the accuracy. Given a question Q and a set of candidates A containing the correct answer A^+, the goal is to rank the correct answer A^+ higher than the $|A| - 1$ candidates that do not contain the right answers.

SUBJ dataset [16]: As a specialized dataset for subjectivity detection task, subjectivity dataset (SUBJ) consists of 1,346 hand-annotated documents drawn from the top 20 webpages retrieved by the Yahoo! search engine in response to 69 real user queries. The annotations of the dataset indicate whether the statements are subjective or objective. In total, the dataset consists of 5,000 subjective and 5,000 objective sentences.

Wikipedia biased statements [8]: This dataset was released by Christoph Hube and Besnik Fetahu. The dataset is constructed with two steps: (1) First they extract POV-tagged statements from Wikipedia, suggesting that the statements are against the "Neutral point of view (NPOV)" principle in Wikipedia. However, the quality of the raw dataset is not satisfying. (2) Then, they utilize crowdsourcing to manually construct ground-truth. Finally, the released dataset consists of 1,843 biased statements, 3,109 neutral ones, 1,843 neutral ones from featured articles in Wikipedia and 1,843 neutral ones from featured articles equipped with same type-balanced distribution in biased statements. In the experiment part, we choose biased statements and featured neutral statements with similar type-balanced distribution.

IMDb dataset [13]: IMDb dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb), labeled as positive or negative, containing substantially more data than previous work. The dataset contains an even number of positive and negative reviews.

Table 1: Dataset summary

Dataset	Dataset size	Label	Task
STS-B	8,628	1	Text similarity task
SNLI	570,152	3	Pair-wise text classification task
QNLI	115,669	2	Pair-wise relevance ranking task
SUBJ	10,000	2	Subjectivity detection task (single sentence classification, main task)
Wikipedia biased statements	3,686	2	Subjectivity detection task (single sentence classification, main task)
IMDb	50,000	2	Single sentence classification

4.2 Implementation details

Our experiments are based on PyTorch implementation of BERT. We utilize the BERT-base uncased model consisting of 12 layer transformer blocks and 12 heads as well as 768 hidden units, 110M parameters in total [5]. As for optimizer, we utilize Adam optimizer [10] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We set the batch size to 32, the number of epochs to 5 and dropout probability to 0.1. The base learning rate is 2e-5, while the maximum and minimum momentum are 0.9 and 0.8, respectively.

4.3 Results

Since our main task is the subjectivity detection task, which is a branch of single sentence classification, the following experiments and results focus on the main task. In order to evaluate the impact of each fine-tuning strategy on subjectivity detection task, three baseline models will be used: (1) Standard BERT-base fine-tuning without applying any strategies. (2) BERT-base model with LSTM network. (3) BERT-base model with Bi-LSTM network.

Moreover, we compare our methods with the current state-of-art models as far as we know for each dataset. For SUBJ dataset, we compare with AdaSent [27]. For Wikipedia biased statement, we compare our model

against the neural-based model with hierarchical attention proposed by Hube et al.[8]. Also, although our main task is subjectivity detection task, we include IMDb dataset in our experiment result, since it is a single sentence classification task just like SUBJ dataset and Wikipedia biased statements. For the IMDb dataset, we compare our model with XLNet [26].

To keep consistency, we report all results by accuracy. The Table 2 shows our experimental results on the three classification tasks.

• Impact of different classifiers upon BERT

First, we investigate the impact of choosing different classifiers upon BERT. In addition to applying a simple softmax classifier, we choose prevalent neural networks LSTM and Bi-LSTM to see whether choosing a more sophisticated method would boost the performance or not. The result in Table 2 shows that choosing a more complex classifier does not improve the performance. Instead, it would rather decrease the accuracy on the three classification tasks, which makes sense since BERT already consists of deep networks as well as sophisticated training strategies. Adopting a more complex classifier is not a compulsory option.

• Impact of layer-wise discriminative fine-tuning

To investigate the influence of each fine-tuning strategy on performance, we further utilize the BERT-base model with the combination with each strategy separately. As for discriminative fine-tuning, the result in Table 2 shows that there was no significant improvement in accuracy.

• Impact of one cycle policy

In applying the one cycle policy, we observe an improvement on the performance. Compared to the basic BERT fine-tuning, utilizing one cycle policy would boost the performance on SUBJ around 0.3% from standard BERT fine-tuning, while 0.6% on Wikipedia biased statement dataset from standard BERT fine-tuning as well. However, one cycle policy does not show improvement on IMDb. Nevertheless, one cycle policy proves to be effective on smaller datasets, suggesting its ability to prevent over-fitting problem.

• Impact of gradual unfreezing

As for the gradual unfreezing strategy, the result shows that applying gradual unfreezing does not help the model to outperform a standard BERT fine-tuning.

• Impact of multi-task learning

(1) For the evaluation of multi-task learning, we first utilize three classification datasets: SUBJ, Wikipedia biased statement and IMDb. The result shows that there is a significant improvement on the three classification datasets, suggesting that BERT and multi-task learning can be complementary. (2) To further investigate the influence of choosing a wider range of tasks, we then utilize four tasks and six datasets completely. Not only the result improves compared to standard BERT-base fine-tuning, but also surpasses the result of only utilizing the three classification tasks. In fact, we achieve the best result both on SUBJ and IMDb datasets with accuracy of 95.23% and 93.07%, demonstrating the effectiveness of multi-task learning with using a wider range of tasks.

• Impact of the combination of MTL and 1 cycle policy

Since we find that one cycle policy and multi-task learning can both be effective on boosting performance with BERT, how is the case utilizing both of them? The experimental result shows that there is a slight improvement on the Wikipedia biased statement dataset, and we achieve the best accuracy 84.05% on it by combining multi-task learning and one cycle policy, surpassing the best result from [8], while there is no sign of outperforming than merely utilizing multi-task learning on SUBJ and IMDb dataset.

Table 2: Accuracy of different strategies with BERT.

Model	SUBJ	Wikipedia biased statement	IMDb
State-of-art			
AdaSent (Zhao et al., 2015)	95.5	/	/
Neural-based model with attention mechanism (Hube et al., 2019)	/	80.8	/
XLNet (Yang et al., 2019)	/	/	96.21
Baseline models			
BERT-base fine-tuning	94.2	81.56	91.51
BERT-base -LSTM	92.94	80.42	87.4
BERT-base -BiLSTM	92.48	80.94	87.46
Fine-tuning strategies			
BERT-base (Discriminative)	94.18	81.61	91.13
BERT-base (1 cycle policy)	94.53	82.17	91.43
BERT-base (Gradual unfreezing)	93.98	81.29	91.47
BERT-base (MTL, 3 datasets)	95.02	83.04	92.86
BERT-base (MTL, 6 datasets)	**95.23**	83.81	**93.07**
BERT-base (MTL, 6 datasets and 1 cycle policy)	95.18	**84.05**	92.98

5. Conclusion

In this paper, we first investigate how to utilize a standard BERT fine-tuning for subjectivity detection task, then compare the performance of different classifiers upon BERT, proving that using BERT can spare the need of complex neural classifiers. In addition, we discuss several

fine-tuning strategies and conduct experiments on classification task. Among our experiments, there is a significant improvement by the combination of one cycle policy and multi-task learning strategy on Wikipedia biased statement dataset, surpassing the best result from [8], while utilizing multi-task learning with 6 datasets and 4 tasks can achieve satisfying results on SUBJ and IMDb datasets. Moreover, experiments prove that choosing a wider range of tasks in multi-task learning can benefit the results more than smaller range of tasks. There are many future areas to explore further, including the structure of information sharing mechanism inside multi-task learning.

References

[1] D. Aleksandrova, F. Lareau, P. Ménard, Multilingual Sentence-Level Bias Detection in Wikipedia, 2019.

[2] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. *NIPS*, 2006.

[3] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," *Machine Learning,* vol. 28, pp. 41–75, 1997.

[4] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval2017 Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proc. 11th Int. Workshop Semantic Eval.*, Aug. 2017, pp. 1–14.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[6] Hube, C., and Fetahu, B. 2018. Detecting biased statements in wikipedia. *In The Web Conference*, 1779–1786. International World Wide Web Conferences Steering Committee.

[7] J. Howard, S. Ruder, "Universal language model fine-tuning for text classification", *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, pp. 328-339, 2018.

[8] Hube, C., and Fetahu, B. 2019. Neural based statement classification for biased language. *In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 195–203. ACM.

[9] M. Karamibekr and A. A. Ghorbani, "Sentence subjectivity analysis in social domains," *in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence(WI-2013),* Atlanta, GA, USA, November.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[11] C. Lin, Y. He, R. Everson, "Sentence subjectivity detection with weakly-supervised learning", *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, pp. 1153-1161, Nov. 2011.

[12] X. Liu, P. He, W. Chen, and J. Gao, ''Multi-task deep neural networks for natural language understanding,'' in *Proc.57th Annu. Meeting Assoc. Comput. Linguistics,* 2019, pp. 4487–4496.

[13] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, ''Learning word vectors for sentiment analysis,'' in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. 2011, pp. 142–150.

[14] A. Montoyo, P. Martínez-Barco, A. Balahur, "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments", *Decision Support Syst.,* vol. 53, no. 4, pp. 675-679, Mar.–Apr. 2012.

[15] M. V. Mäntylä, D. Graziotin, M. Kuutila, "The. evolution of sentiment analysis—A review of research topics venues and top cited papers", *Comput. Sci. Rev.,* vol. 27, pp. 16-32, Feb. 2018.

[16] B. Pang, L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In Proceedings of ACL 2004.*

[17] Recasens M., Danescu-Niculescu-Mizil C. and. Jurafsky D.: Linguistic Models for Analyzing and Detecting Biased Language. *Proceedings of ACL* (2013).

[18] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000 + Questions for Machine Comprehension of Text. In *EMNLP*, 2016.

[19] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," 2017. [Online]. Available: http://arxiv.org/abs/1706.05098.

[20] Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, 2018, [online] Available: https://blog.openai.com/language-unsupervised/.

[21] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642, 2015.

[22] L. N. Smith. Cyclical learning rates for training neural networks. In *WACV*, 2017.

[23] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018

[24] C. Sun, X. Qiu, Y. Xu, X. Huang, "How to fine-tune BERT for text classification?", *arXiv:1905.05583,* May 2019, [online] Available: https://arxiv.org/abs/1905.05583.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention Is All You Need", *CoRR,* vol. abs/1706.03762, 2017.

[26] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, ''XLNet: Generalized autoregressive pretraining for language understanding,'' 2019, *arXiv:1906.08237*. [Online]. Available: https://arxiv.org/abs/1906.08237

[27] H. Zhao, Z. Lu, and P. Poupart. (2015). ''Self-adaptive hierarchical sentence model.'' [Online]. Available: https://arxiv.org/abs/1504.05070

[28] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114,* 2017.

[29] L. Zhang, S. Wang, B. Liu, "Deep learning for sentiment analysis: A survey", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, pp. 25, Mar. 2018.