PTE: Sentiment-aware Pretext Task Enhanced Fine-tuning for Sentence-level Sentiment Analysis

Anonymous EMNLP submission

Abstract

Pre-trained models have achieved huge success in many NLP tasks, however, most of the existing general pre-trained models do not pay special attention to sentiment knowledge, which is important to the sentiment analysis task. To solve this problem, we propose sentiment-aware pretext tasks to enhance the pre-trained models (PTE) by incorporating external sentiment knowledge. More specifically, we study sentence-level sentiment analysis and, correspondingly, propose two sentiment-aware pretext tasks named Sentiment Word Cloze (SWC) and Conditional Sentiment Prediction (CSP). The SWC task learns to recover sentiment words ascription given the overall sentiment polarity of a sentence. On the contrary, CSP predicts sentence sentiment polarity given the word sentiment polarity as prior knowledge. We conduct experiments on eight datasets and the results demonstrate that our approach consistently outperforms baseline pre-trained models for sentence-level sentiment analysis. The code and data are released at xxx (anonymous during the review).

1 Introduction

Recently, pre-trained language models (PTMs) have achieved state-of-the-art performance on many downstream NLP tasks[1]. Generally, these PTMs are pre-trained from large-scale unlabeled general corpora by various well-designed pre-training tasks. To capture different semantic information, different pre-training tasks are designed in these PTMs, e.g., BERT proposed masked language model (MLM) and next sentence prediction (NSP); ALBERT (Lan et al., 2019) proposed sentence order prediction (SOP); ERNIE (Sun et al., 2020) proposed knowledge masking and sentence reordering, and so on.

In addition to the pre-training tasks, many pretext tasks, based on self-supervised learning (Liu et al., 2020), are proposed to enhance pre-trained models with useful task-specific information during the fine-tuning stage. For the sentiment analysis task, there are also some pretext tasks proposed, such as sentiment word prediction, word polarity prediction and aspect-sentiment pairs prediction (Tian et al., 2020), part-of-speech (POS) tag prediction (Gu et al., 2020), etc. They are usually designed to incorporate external knowledge such as the word sentiment polarity. However, we argue that previous methods are not effective and efficient enough. Firstly, previous methods try to reconstruct the masked sentiment word with MLM calculated on the whole vocabulary, which is not efficient. Secondly, most of the the existing pretext tasks try to predict word sentiment polarity according to a sentiment vocabulary, which is not always accurate. Therefore, the word sentiment polarity is not suitable to be training labels alone.

To alleviate the above problem, in this paper, we propose two novel pretext tasks to enhance the PTMs for sentiment analysis, we call it *Pretext Task Enhanced* PTMs (PTE). Our method starts from building the sentiment vocabulary from public resources and recognizing all the sentiment words in the input sentence. Then, two pretext tasks are proposed to incorporate the word-level sentiment information into PTMs during fine-tuning stage. Different from previous methods, our pretext tasks focus to model the relationship between the incorporated word sentiment polarity and the original sentence sentiment polarity, rather than the word-level information solely. Specifically, the first task is Sentiment Word Cloze (SWC), which recovers the masked sentiment words from some sampled sentiment word candidates, given the sentence sentiment polarity. This task is to model the influence of the sentence polarity on its sentiment words. The second task is Conditional Sentiment Prediction (CSP), which predicts the sentiment polarity of a sentence, given the sentiment polarity of the

[1] https://gluebenchmark.com/leaderboard

word in it. This task is to model the influence of the word polarity on its ascribed sentence polarity. Moreover, we take the proposed pretext tasks as a joint-label classification problem and explore two ways to jointly leverage the incorporated word-level labels and the original sentence-level labels. With our proposed two pretext tasks, PTMs are enhanced to learn better semantic representation for sentence-level sentiment analysis. Our contributions are as follows.

- We propose two novel pretext tasks to learn better sentiment-aware PTMs for sentence-level sentiment analysis, enhancing PTMs via modelling the relationship between the incorporated word-level sentiment information and the original sentence-level one.
- We propose two ways to jointly optimize multiple kinds of labels, which describe the input from multiple perspectives.
- Extensive experimental results on 5 English and 3 Chinese datasets show that PTE significantly outperforms the strong pre-trained models on sentence-level sentiment analysis.

2 Related Work

PTMs and Pre-training Tasks. Pre-trained Models have achieved remarkable improvement in many NLP tasks, and many variants of PTMs have been proposed. For example, GPT, GPT-2 and GPT-3 (Radford et al., a,b; Brown et al., 2020), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and ALBERT (Lan et al., 2019), ERNIE (Sun et al., 2020), BART (Lewis et al., 2020) and RoBERTa (Liu et al., 2019). Most PTMs are pre-trained with pre-training tasks with large-scale unlabeled corpora. The pre-training tasks are summarized in the first block in Table 1.

External Knowledge for Sentiment Analysis. Before PTMs, there have been some work about incorporating external sentiment resources into neural models like Long Short-Term Memory (LSTM) (Graves et al., 2013) or Convolutional Neural Networks (CNN) (Kim, 2014). They are enhanced by sentiment strength, sentiment lexicons, negation words and intensity words (Bao et al., 2019; Qian et al., 2017; Teng et al., 2016; Zou et al., 2018; Shin et al., 2017), etc.

Post-training with Pretext Tasks for Sentiment Analysis. Post-training refers to continuing pre-training with domain-specific or task-specific corpora (Xu et al., 2019), which can improve the downstream task performance (Gururangan et al., 2020). To realize it, for sentiment analysis, some pretext tasks are proposed to incorporate external knowledge into PTMs. For example, SKEP (Tian et al., 2020) leverages sentiment word prediction, word polarity prediction and aspect-sentiment pair prediction to enhance PTMs with sentiment knowledge. SentiLARE (Ke et al., 2020) uses sentiment word prediction, word polarity prediction and word part-of-speech (POS) tag prediction and joint prediction tasks. SENTIX (Zhou et al., 2020) designs sentiment word prediction, word polarity prediction, emoticon prediction and rating prediction tasks. The second block in Table 1 shows pretext tasks proposed to post-train PTMs.

Model	Pre-training/Pretext Tasks
BERT	MLM and NSP
ALBERT	sentence order prediction
ERNIE	knowledge mask
	sentence reordering
BART	token mask/deletion
	sentence permutation
SKEP	sentiment word prediction
	word polarity prediction
SentiLARE	word, and its polarity/POS
SENTIX	sentiment word, emotion
	word polarity, rating
PTE (our)	sentiment word cloze
	conditional sentiment prediction

Table 1: An overview of recent self-supervised tasks. The first block is pre-training tasks during pre-training and the second block is pretext tasks during post-training or fine-tuning.

However, our work is different from the above. First, in the Sentiment Word Cloze task, instead of reconstructing the masked word, we predict whether a given sentiment word belong to the input sentence or not. In other words, we recover the masked sentiment words from the sentiment word candidates instead of the whole vocabulary. This reduces the amount of classification parameters and the difficulty of the pretext task. Second, in the Conditional Sentiment Prediction task, instead of predicting word sentiment polarity itself, we treat it as prior knowledge to assist predicting sentence sentiment polarity. This avoids the inaccuracy of word sentiment polarity. Third, instead of incor-

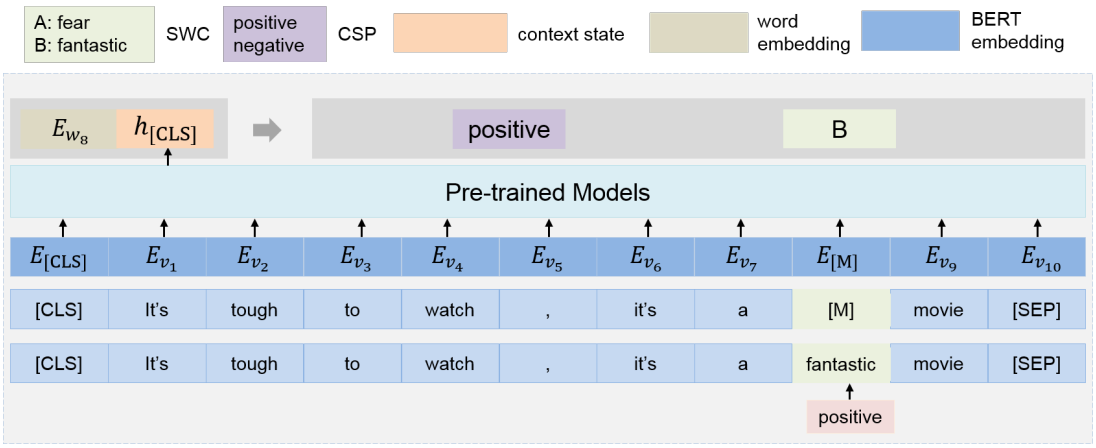

Figure 1: Overview of PTE. Firstly, at the bottom of this figure, sentiment word `fantastic` and its sentiment polarity `positive` are recognized by external sentiment vocabulary, and the word `fantastic` is replaced with `[M]`. The input is then tokenized into subwords and input into PTMs to obtain context state $h_{[\text{CLS}]}$, meanwhile, the embedding E_{w_8} of the masked sentiment word is extracted. Finally, for the SWC task, the masked sentiment word `fantastic` and the sampled sentiment word `fear` are treated as candidates. For the CSP task, the masked sentiment word polarity `positive` is treated as prior knowledge to predict sentence sentiment polarity. Note that, the pink rectangle and purple rectangle refer to the sentiment polarity of word and sentence, respectively.

porating in the post-training stage, our proposed two pretext tasks are conducted in the fine-tuning stage, which brings nearly no difference in the final performance but is easier to implement.

3 Methodology

Figure 1 illustrates the framework of PTE, which extends general pre-trained models with our proposed Sentiment Word Cloze (Section 3.1) and Conditional Sentiment Prediction (Section 3.2) tasks. They are proposed to model the relationship between the word polarity and the sentence polarity. SWC selects sentiment words that belongs to the input sentence, given the polarity of it, while CSP predicts sentence sentiment polarity based on the sentiment polarity of words in it. In the subsequent subsections, we will introduce the two proposed pretext tasks. For convenience, we first give some notations used in the following subsections.

Formally, let $L = \{l_1, l_2, \cdots, l_M\}$ denote the sentiment vocabulary with size M, and $S = \{w_1, w_2^l, \cdots, w_K^l, \cdots, w_N\}$ denote an input sentence of length N which includes $0 \leq K \leq N$ sentiment words. Given sentiment polarity sets $C = \{C_i\}_{i=1}^T$, where T is the number of polarity labels. $P_S, P_{w_k^l} \in C$ represent the sentiment polarity of sentence S and word w_k^l, respectively. Note that for a sentiment word, it only has three kinds of polarities, "positive", "neutral" and "negative". For a sentence, it has up to five kinds of polarities. $Y_{w,S} \in \{0, 1\}$ represents the ascription relationship between word w and sentence S, where $Y_{w,S} = 1$ means that w belongs to S. $E \in \mathbb{R}^{V \times d}$ is the embedding table, where V is the task vocabulary size and d is the dimension of embeddings. The goal of sentiment analysis is to predict the overall sentiment polarity P_S of S.

The main task of sentence-level sentiment analysis is to predict the sentiment label P_S given the input sentence S. Firstly, the input S is passed through PTMs to get the context state $h^S_{[\text{CLS}]}$. Then the context state is fed into a linear layer and a Softmax layer to get the probability $\hat{P}_S$ of each sentiment label, i.e., $\hat{P}_S = \text{Softmax}(W_1 h^S_{[\text{CLS}]} + b_1)$, where W_1 and b_1 are the model parameters. The loss function of the main task is the cross-entropy:

$$\mathcal{L}_{main} = -\frac{1}{|C|} \sum_{i \in C} P_S^i \cdot \log(\hat{P}_S^i) \qquad (1)$$

3.1 Sentiment Word Cloze

The purpose of Sentiment Word Cloze (SWC) is to promote the model to learn the influence of the sentence sentiment polarity on the sentiment word in the sentence. Given a training sample (S, P_S), we first recognize all the sentiment words in S according to the sentiment vocabulary L. Then, we choose one of them as w_k^l, and record its sentiment polarity as $P_{w_k^l}$. We mask w_k^l in S with a special token $[M]$ to get the corrupted sentence S'. Meanwhile, we randomly sample one sentiment word

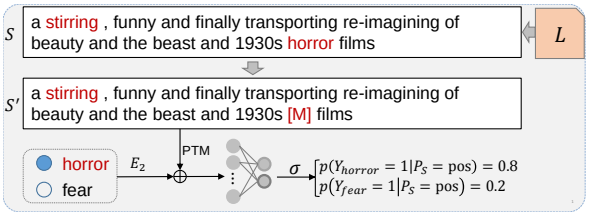

Figure 2: An example of SWC task.

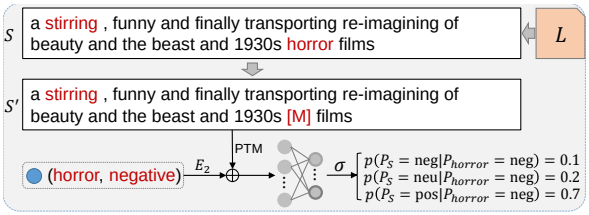

Figure 3: An example of CSP task.

from L as w_j^l and record its sentiment polarity as $P_{w_j^l}$ which will be used in Section 3.3. Next, S' is fed into PTMs to get the context state $h^{S'}_{[\text{CLS}]}$. Meanwhile, we extract the embeddings of the words $P_{w_k^l}$ and $P_{w_j^l}$ as e. Then a linear layer and a Softmax layer is used to compute each candidate's probability $\hat{Y}_{w^l,S} = \text{Softmax}(W_2(h^{S'}_{[\text{CLS}]} + e) + b_2)$, where W_2, b_2 is model parameters. The loss function is defined as the cross-entropy between the predicted probability $\hat{Y}_{w^l,S}$ and the truth word ascription label $Y_{w^l,S}$, i.e.,

$$\mathcal{L}_{SWC} = -\frac{1}{|\Omega|}\sum_{i\in\Omega} Y_{w_i^l,S} \cdot \log(\hat{Y}_{w_i^l,S}) \quad (2)$$

where Ω means all the classification labels. The final loss is the sum of the main task (sentiment analysis) loss $\mathcal{L}_{main}$ and the SWC loss $\mathcal{L}_{SWC}$ weighted by α, i.e., $\mathcal{L} = \mathcal{L}_{main} + \alpha\mathcal{L}_{SWC}$.

Figure 2 gives an example of the procedure of SWC. In this example, "stirring" and "horror" are recognized as sentiment words, and the masked sentiment word is "horror" and the sampled sentiment word is "fear". After the mask operation, the corrupted sentence S' is input into PTMs to get the context state, which will be passed through a linear layer together with the embeddings of the masked and sampled words. Finally, the probability of whether these two words belong to S is obtained with a Softmax layer.

3.2 Conditional Sentiment Prediction

Conditional Sentiment Prediction (CSP) aims to learn the influence of the sentiment polarity of a word on its ascribed sentence polarity. We argue that with the word sentiment polarity given as prior knowledge, PTMs could learn more sentiment information explicitly, thereby yielding better sentiment analysis performance. Given a training sample S, P_S, similar to SWC, we first choose one sentiment word w_k^l from all sentiment words in S recognized with L, meanwhile recording its sentiment polarity $P_{w_k^l}$, which will be used according to Section 3.3. Next we mask the word w_k^l with $[M]$ to get the corrupted input sentence S', which is fed into PTMs to get the context state $h^{S'}_{\text{CLS}}$. The context state and the embedding of w_k^l, notated as e, are passed through a linear layer and a Softmax layer to predict the sentence polarity. i.e., $\hat{P}_S = \text{Softmax}(W_3(h^{S'}_{[\text{CLS}]}+e)+b_3)$, where W_3, b_3 is model parameters. The loss function of CSP is the cross-entropy between the predicted probability and the ground-truth sentence label, i.e.,

$$\mathcal{L}_{CSP} = -\frac{1}{|\Omega|}\sum_{i\in\Omega} P_S^i \cdot \log(\hat{P}_S^i) \quad (3)$$

where Ω means all the classification labels. The final loss is the sum of $\mathcal{L}_{main}$ and the CSP loss $\mathcal{L}_{CSP}$ weighted by β, i.e., $\mathcal{L} = \mathcal{L}_{main} + \beta\mathcal{L}_{CSP}$.

Figure 3 gives an example of the procedure of the Conditional Sentiment Prediction task. In this example, the chosen sentiment word is "horror" and its sentiment polarity is "negative". After masking, the corrupted sentence is input into PTMs to get the context state, which will be passed through a linear layer together with the embedding of "horror". Finally, the probability of the sentence polarity is obtained with a Softmax layer.

3.3 Joint-label Classification

Different from traditional sentence-level sentiment analysis, our proposed method incorporates the external knowledge of sentiment words into the training process, which means we need to consider the sentence polarity and the word polarity together as multiple labels. Intuitively, multiple kinds of labels can describe the input sentence from different perspectives, encouraging the model to leverage different useful information at the same time. For the SWC task, in addition to the sentence polarity label P_S, we also need to consider the word ascription label Y, which describes the relationship between the sentence and the sentiment word in it. Correspondingly, for the CSP task, in addition to the global sentiment P_S, we also need to leverage

the word sentiment polarity $P_{w_k^l}$, which describes the local sentiment. To process the multiple labels, we propose two kinds of joint-label classification methods. The first one is joint inference (JI), which models the joint distribution of the multiple-labels. This method essentially treats multiple-labels as a single label defined on the Cartesian product of different kinds of label. The second way is aggregated inference (AI) motivated by Lee et al. (2020), which models the conditional distribution of multiple-labels. This method essentially predicts one kind of label label with other kinds of label as prior knowledge.

With joint-label classification, we need to modify the calculation of the predicted probabilities in Eq.(2) and (3).

Joint inference. For SWC, given the linear representation $X = W_2(h^{S'}_{[\text{CLS}]} + e) + b_2$, we need to predict the joint distribution of the word ascription label Y and the sentence polarity P_S through a Softmax layer, i.e., $p(Y, P_S|X) \in \mathbb{R}^{2\times|C|}$, where 2 means the number of Y's label ($\{0, 1\}$) and $|C|$ means the number of P_S's labels. We use $p(Y, P_S|X)$ to calculate the loss in Eq.(2), meaning $|\Omega| = 2 \times |C|$.

For CSP, given the linear representation $X = W_3(h^{S'}_{[\text{CLS}]} + e) + b_3$ and its sentence representation Z, we need to predict the joint distribution of the word polarity $P_{w_k^l}$ and the sentence polarity P_S through a Softmax layer, i.e., $p(P_{w_k^l}, P_S|X) \in \mathbb{R}^{2\times|C|}$, where 2 means the number of $P_{w_k^l}$'s label ($\{$"*positive*", "*negative*"$\}$) and $|C|$ means the number of P_S's labels. We use $p(P_{w_k^l}, P_S|X)$ to calculate the loss in Eq.(3), meaning $|\Omega| = 2 \times |C|$.

Aggregated inference. For SWC, given the linear representation X, we want to predict the word ascription label Y with the sentence polarity P_S as the prior knowledge, i.e., $p(Y|X, P_S) \in \mathbb{R}^2$, meaning $|\Omega| = 2$ in Eq.(2). To get this, we simply choose the according two logits from $p(Y, P_S|X)$ followed by normalization.

Similarly, for CSP, the conditional probability of sentence sentiment polarity P_S given the word sentiment polarity $P_{w_k^l}$ is $p(P_S|X, P_{w_k^l}) \in \mathbb{R}^{|C|}$, meaning $|\Omega| = |C|$ in Eq.(3). To get this, we simply choose the according $|C|$ logits from $p(P_{w_k^l}, P_S|X)$ followed by normalization.

Dataset	#train	#val	#test	#W	#C
MR	5,330	5,331	–	22	2
SST2	6,920	872	1821	20	2
Airline	4.8K	4,880	4,880	18	3
SEval	50.3K	12.1K	–	19	3
SST5	8,544	1,101	2,210	20	5
ChnSC	5K	1,466	1,299	128	2
MSS	6,228	11.6K	22.3K	22	3
DMSC	66.9K	8,368	8,370	49	5

Table 2: Datasets used in the experiment. #train, #W and #C denote the number of training samples, classes and averaged sentence length, respectively.

4 Experiment

4.1 Preprocessing

4.1.1 Datasets

A variety of English and Chinese sentence-level sentiment analysis datasets are used for the experiment, as shown in Table 2. The English datasets include Movie Review (MR) (Pang and Lee, 2005), Stanford Sentiment Treebank (SST2 and SST5) (Socher et al., 2013), SemEval-2017 Task 4 Subtask A (SEval) (Rosenthal et al., 2017), Twitter US Airline Sentiment (Airline) [2]. The Chinese datasets include ChnSentiCorp (ChnSC) (Song-bo), Douban Movie Short Comments Dataset V2 (DMSC) [3], Medical Service Sentiment (MSS). For MR, we randomly split it into training and validation sets. For SEval, we only use its training set and validation set for simplicity. For Airline and DMSC, we randomly choose 80%, 10%, 10% samples from it for training, validation and test, respectively. The model performance is evaluated with the accuracy (ACC).

4.1.2 Sentiment Vocabulary

To build the English sentiment vocabulary, we select the words with sentiment scores greater than 0.5 from SentiWordNet 3.0 (Baccianella et al., 2010) as sentiment words, which contains 6,156 sentiment words with positive or negative polarity. To build the Chinese sentiment vocabulary, we merge two online resources HowNet Sentiment Dictionary and Tsinghua Sentiment Dictionary[4]

[2] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[3] https://www.kaggle.com/utmhikari/doubanmovieshortcomments

[4] https://github.com/Shimon-Guo/chinese_sentiment_dictionary

Dataset	In SWN	In HT
MR	79.23%	–
SST2	76.40%	–
Airline	59.22%	–
SEval	57.06%	–
SST5	72.55%	–
ChnSC	–	98.06%
MSS	–	68.34%
DMSC	–	80.92%

Table 3: The percentage of sentences containing sentiment words. SWN is the English sentiment vocabulary and HT is the Chinese sentiment vocabulary.

into one vocabulary, which contains 19,186 sentiment words with positive or negative polarity. Table 3 shows the coverage of the vocabulary in each dataset. The coverage is defined as the percentage of training samples containing the defined sentiment words.

4.2 Implementation Details

We implement our model based on *HuggingFace's Transformers*[5]. For all PTMs, we use their base versions. The batch size is set to 16 for ChnSC, 64 for DMSC, and 32 for other datasets. The learning rate is 2e-5 for XLNet and RoBERTa, and 5e-5 for BERT . The input format and output format are consistent with each PTM. Meanwhile, the input sequence length is set to 50 for MR, 32 for SEval, 350 for ChnSC, 150 for DMSC and 128 for other datasets, so that more than 90% of the samples are used. Other hyper-parameters are consistent with the codes of *HuggingFace's Transformers*. The loss balance weight α and β are searched from $\{0.01, 0.1, 0.5, 0.8, 1.0\}$. We post-train or fine-tune each model for 3 epochs, and the best checkpoints on the development set are used for inference. For each dataset, we run 8 times with different seeds and the average results are reported. Task vocabulary is constructed by splitting sentences into words and characters for English datasets and Chinese datasets, respectively.

4.3 Baselines

To demonstrate the effectiveness of the proposed method for sentence-level sentiment analysis, we compare our method with three types of competitive baselines, including pre-trained models (PTMs), post-trained PTMs by task-specific data and PTMs enhanced by recent proposed pretext tasks for sentiment analysis.

Pre-trained Models. We use the base version of BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019) as our baselines, which are the most popular PTMs.

PTMs with Task-specific Post-training. We continue pre-training the released BERT with MLM on the training set of sentiment analysis datasets, denoted as BERT+PT.

PTMs Enhanced by Pretext Tasks. We also adopt some methods focusing on pretext tasks, which leverage external knowledge, e.g., sentiment vocabulary or POS, to enhance PTMs. We choose two baseline methods, i.e., SentiLARE (Ke et al., 2020) and SENTIX (Zhou et al., 2020). Both of them design the sentiment word prediction task and the word polarity prediction task. The details are listed in Table 1 in Section 2.

4.4 Overall Results

Table 4 and Table 5 list the results of all the baselines and our method (denoted by *). Among the results, "+PTE" means we train the model with the loss $\mathcal{L} = \mathcal{L}_{main} + \alpha\mathcal{L}_{SWC} + \beta\mathcal{L}_{CPS}$."+SWC" means $\beta = 0$ while "+CPS" means $\alpha = 0$.

4.4.1 Comparison with PTMs

From Table 4, when comparing our method PTE, i.e., BERT+PTE, XLNet+PTE and RoBERTa+PTE, with baseline PTMs, i.e., BERT, XLNet and RoBERTa, we find that PTE significantly outperforms baseline PTMs on nearly all datasets, especially on the SST5 dataset, with the improvement of 0.92 to 1.42 points.

We also list the results of SWC solely, i.e., BERT+SWC, XLNet+SWC and RoBERTa+SWC in Table 4. From the table, we find our SWC task outperforms the baseline PTMs on nearly all datasets with up to 1.12 and 0.43 improvement on the English datasets and Chinese datasets respectively. The results verify that the SWC task indeed helps learn useful representations for the main task.

Similarly, we also report the results of PTMs enhanced with only CSP task. We see that PTMs enhanced with CSP perform better than all the baseline PTMs on nearly all the datasets. Specifically, CSP yields up to 0.92 and 0.32 improvement on the English and Chinese datasets respectively, which demonstrates predicting sentence sentiment polarity with word sentiment polarity as prior knowledge helps learning better semantic representations.

[5] https://github.com/huggingface/transformers

Model	MR	SST2	Airline	SEval	SST5	ChnSC	MSS	DMSC
BERT	86.38	91.23	83.32	67.50	53.58	90.33	89.54	57.93
+PT	**86.50**	91.14	83.58	67.58	53.36	90.60	89.51	58.16
+SWC*	86.36	91.35	83.46	67.72	54.14	**90.76**	**89.63**	**58.22**
+CSP*	86.41	91.34	**83.64**	67.81	**54.15**	90.72	89.52	58.06
+PTE*	86.37	**91.42**	83.48	**67.94**	54.00	90.67	89.58	58.17
XLNet	88.04	92.08	84.13	67.80	54.28	–	–	–
+SWC*	88.30	92.90	84.40	67.76	55.40	–	–	–
+CSP*	**88.40**	**92.90**	84.30	67.84	55.20	–	–	–
+PTE*	88.30	92.86	**84.40**	**67.88**	**55.70**	–	–	–
RoBERTa	88.15	93.96	85.09	69.96	56.96	91.15	89.68	58.35
+SWC*	88.20	94.14	85.15	70.17	57.23	91.21	89.85	**58.48**
+CSP*	**88.41**	94.19	85.16	70.15	57.45	91.36	**90.00**	58.37
+PTE*	88.30	**94.21**	**85.20**	**70.17**	**57.60**	**91.36**	89.95	58.42

Table 4: Overall results. * refers to our method.

When comparing all the methods together, we observe that 1) in terms of English datasets, the best improvement is achieved on SST5, followed by Airline and SEval and then SST2 and MR. We argue that PTE is more useful to sentiment analysis with more classes. 2) The improvement on the Chinese datasets is less than that on the English datasets. This may result from the different implementation of masking in the two datasets. For the English datasets, the masked sentiment word looks up its embedding based on the word-level task vocabulary. However, for the Chinese datasets, we use the character-level task vocabulary to avoid word segmentation. The maximum length of masked sentiment words is set to four to fix the input length of PTMs. For the masked Chinese word less than four characters, we pad it with `[PAD]`, and then obtain its embedding e by summing up the embedding of each character. We think this pad operation could introduce noise, which may decrease the improvement of PTE, mainly CSP, on the Chinese datasets. 3) PTE does not always outperform SWC or CSP used alone. This is probably because SWC and CSP may compete with each other. As mentioned in (Bingel and Søgaard, 2017), multiple tasks may promote each other or compete with each other (negative learning). To solve this problem, better strategies to balance the weights of each task or more matching pretext tasks are worth exploring.

4.4.2 Comparison with Post-training

To further demonstrate the effectiveness of PTE, we choose another baseline, i.e., post-training on the training set of each dataset with MLM, given PTMs.

Model	SST2	SST5
SentiLARE$_{SW}$	–	57.31*
RoBERTa+SWC	–	**57.42**
SentiLARE$_{WP}$	–	56.91*
RoBERTa+CSP	–	**57.65**
SENTIX	92.26*	54.34*
BERT+PTE	92.15	**54.62**

Table 5: Comparisons with sentiment word prediction and word polarity prediction task. * means the results are cited from the original paper. – means the model does not report results on this dataset.

This baseline is denoted as "+PT" in Table 4. From the table, we see that BERT+PT only improves the performance with an average of 0.08 points compared with the baseline BERT, which is probably because the PTM has contained much semantic information learned by MLM and continuing training with the same task gains little. However, we see that our methods BERT+SWC, BERT+CSP and BERT+PTE improve BERT with an average of 0.22, 0.23 and 0.23 points, which are better than BERT+PT. The result demonstrates that continuing training with more pretext tasks can indeed boost the performance of general PTMs, and our proposed pretext tasks can enhance the PTMs with more sentiment information to help the main task.

4.4.3 Comparison with other Pretext Tasks

We also compare our pretext tasks with previous pretext tasks for sentiment analysis. In order to verify the superiority of SWC over the sentiment word prediction task, we compare RoBERTa+SWC

BERT	+(a)	+(b)	+(c)	+(d)
SST2	-0.11	**0.37**	-0.18	0.28
SST5	-0.21	-0.08	0.02	**0.45**
XLNet	**+(a)**	**+(b)**	**+(c)**	**+(d)**
SST2	**0.74**	**0.74**	0.63	0.44
SST5	0.37	0.62	**1.14**	0.90
RoBERTa	**+(a)**	**+(b)**	**+(c)**	**+(d)**
SST2	-0.04	**0.41**	-0.11	0.10
SST5	-0.27	-0.39	**0.01**	-0.02

Table 6: Comparisons of SWC options. − denotes performance drop. $\alpha = 1.0$ to focus on the SWC task.

Model	MR	SST2	SEval	SST5
$\text{BERT}_{\text{A+JI}}$	86.27	91.00	**67.99**	53.20
$\text{BERT}_{\text{A+AI}}$	**86.32**	**91.12**	67.68	**53.37**
$\text{BERT}_{\text{B+JI}}$	86.19	90.91	90.41	53.74
$\text{BERT}_{\text{B+AI}}$	**86.35**	**91.14**	**90.67**	**53.82**
$\text{XLNet}_{\text{A+JI}}$	**88.22**	**92.91**	**67.80**	**55.35**
$\text{XLNet}_{\text{A+AI}}$	88.17	92.82	67.62	54.65
$\text{XLNet}_{\text{B+JI}}$	87.86	**93.00**	**67.84**	54.85
$\text{XLNet}_{\text{B+AI}}$	**88.14**	92.80	67.66	**55.09**

Table 7: Comparison of joint inference (JI) and aggregated inference (AI) in joint-label classification. A refers to SWC and B refers to CSP.

with $\text{SentiLARE}_{\text{SW}}$, which uses sentiment word prediction task and late supervision. Meanwhile, to verify the superiority of CSP over the word polarity prediction, we conduct experiments to compare RoBERTa+CSP with $\text{SentiLARE}_{\text{WP}}$, which uses the word-level polarity prediction task and the POS prediction task. For a fair comparison, we use their reported parameters. The results are reported on two common datasets, i.e., SST2 and SST5, as shown in Table 5. From the table, we see that both RoBERTa+SWC and RoBERTa+CSP outperform the compared baselines, with the improvement of 0.11 and 0.54 points. Besides, instead of comparing the single task "SWC" or "CPS" with SENTIX, we compare PTE with SENTIX, since SENTIX does not provide the ablation study of each component. The comparison results show that PTE is 0.11 points lower on the SST2 dataset, but 0.28 points higher on the SST5 dataset. Note that, SENTIX uses the emotion prediction and rating prediction tasks in addition to the sentiment word prediction and the word sentiment polarity prediction tasks, which leverages more external resources than us. Even so, we still obtain the comparable results.

4.5 Ablation Study

4.5.1 Analysis on the Sampling in SWC

To demonstrate the impacts of different sampling ways of candidates in the Sentiment Word Cloze (SWC) task, we design four kinds of sampling methods. (a) **sentiment-random.** Randomly sample candidates from sentiment words of the input and task vocabulary. (b) **random-random.** Randomly sample candidates from the input and task vocabulary. (c) **sentiment-same.** Randomly sample candidates from sentiment words of the input and sentiment vocabulary. Two candidates have the same sentiment polarity. (d) **sentiment-opposite.** The same as (c) except that two candidates have the opposite sentiment polarity. Table 6 compares these variants of sampling method, due to the space limit, we only report the results on SST2 and SST5. We can see that, firstly, the sampling strategy has a greater impact on XLNet while a smaller impact on RoBERTa. Secondly, **random-random** is more suitable on SST2, while **sentiment-same** is more recommended for SST5.

4.5.2 Analysis on Joint-label Classification

In terms of joint-label classification, we compare the joint inference and aggregated inference. The result is shown in Table 7. For both pretext tasks, the aggregated inference is better than the joint inference in most cases on the BERT model, while joint inference has obvious advantages over aggregated inference on the XLNet model, and there is no obvious difference on the RoBERTa model. To conclude, there is no significant difference between the two ways of joint-label classification. Overall, the joint inference is better for SWC while the aggregated inference is recommended for CSP.

5 Conclusion

In this paper, we propose a sentiment-aware pretext tasks enhanced model called PTE for sentence-level sentiment analysis. Sentiment Word Cloze task and Conditional Sentiment Prediction task are designed to incorporated sentiment lexicons for pre-trained model during fine-tuning. In addition, joint-label classification is designed to process multi-label in one loss function. Experiments show that PTE outperforms strong baselines on various sentiment analysis datasets, verifying the necessity of utilizing pretext tasks.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

Lingxian Bao, Patrik Lambert, and Toni Badia. 2019. Attention and lexicon regularized lstm for aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 253–259.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2020. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. 2020. Self-supervised label augmentation via input transformations. In *37th International Conference on Machine Learning, ICML 2020*. ICML 2020 committee.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. 2017. Linguistically regularized lstm for sentiment classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1679–1689.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. a. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. b. Language models are unsupervised multitask learners.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Bonggun Shin, Timothy Lee, and Jinho D Choi. 2017. Lexicon integrated cnn models with attention for sentiment analysis. *EMNLP 2017*, page 149.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

TAN Song-bo. Chnsenticorp.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Zhiyang Teng, Duy-Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1629–1638.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, et al. 2020. Skep: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, page 5754–5764.

Jie Zhou, Junfeng Tian, Rui Wang, Yuanbin Wu, Wenming Xiao, and Liang He. 2020. Sentix: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 568–579.

Yicheng Zou, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2018. A lexicon-based supervised attention model for neural sentiment analysis. In *Proceedings of the 27th international conference on computational linguistics*, pages 868–877.