

Sentiment-Aware Word and Sentence Level Pre-training for Sentiment Analysis

Shuai Fan¹, Chen Lin^{1*}, Haonan Li^{2†}, Zhenghao Lin^{1†}, Jinsong Su¹, Hang Zhang³,
Yeyun Gong⁴, Jian Guo³, Nan Duan⁴

¹ School of Informatics, Xiamen University, China

² The University of Melbourne, Australia

³ IDEA Research, China

⁴ Microsoft Research Asia

Abstract

Most existing pre-trained language representation models (PLMs) are sub-optimal in sentiment analysis tasks, as they capture the sentiment information from word-level while under-considering sentence-level information. In this paper, we propose SentiWSP, a novel **Sentiment**-aware pre-trained language model with combined **Word**-level and **Sentence**-level **Pre**-training tasks. The word level pre-training task detects replaced sentiment words, via a generator-discriminator framework, to enhance the PLM’s knowledge about sentiment words. The sentence level pre-training task further strengthens the discriminator via a contrastive learning framework, with similar sentences as negative samples, to encode sentiments in a sentence. Extensive experimental results show that SentiWSP achieves new state-of-the-art performance on various sentence-level and aspect-level sentiment classification benchmarks. We have made our code and model publicly available at <https://github.com/XMUDM/SentiWSP>.

1 Introduction

Sentiment analysis plays a fundamental role in natural language processing (NLP) and powers a broad spectrum of important business applications such as marketing (HaCohen-Kerner, 2019) and campaign monitoring (Sandoval-Almazán and Valle-Cruz, 2020). Two typical sentiment analysis tasks are sentence-level sentiment classification (Xu et al., 2019; Yin et al., 2020; Tang et al., 2022) and aspect-level sentiment classification (Li et al., 2021b).

Recently, pre-trained language representation models (PLMs) such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018, 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) have brought impressive performance improvements in many NLP problems,

including sentiment analysis. PLMs learn a robust encoder on a large unlabeled corpus, through carefully designed pre-training tasks, such as masked token or next sentence prediction.

Despite their progress, the application of general purposed PLMs in sentiment analysis is limited, because they fail to distinguish the importance of different words to a specific task. For example, it is shown in (Kassner and Schütze, 2020) that general purposed PLMs have difficulties dealing with contradictory sentiment words or negation expressions, which are critical in sentiment analysis. To address this problem, recent sentiment-aware PLMs introduce word-level sentiment information, such as token sentiments and emoticons (Zhou et al., 2020), aspect word (Tian et al., 2020), word-level linguistic knowledge (Ke et al., 2020), and implicit sentiment-knowledge information (Li et al., 2021b). These word-level pre-training tasks, e.g., sentiment word prediction and word polarity prediction, mainly learn from the masked words and are not efficient to capture word-level information for all input words. Furthermore, sentiment expressed in a sentence is beyond the simple aggregation of word-level sentiments. However, general purposed PLMs and existing sentiment-aware PLMs under-consider sentence-level sentiment information.

In this paper, we propose a novel sentiment-aware pre-trained language model called SentiWSP, to combine word-level pre-training and sentence-level pre-training. Inspired by ELECTRA (Clark et al., 2020), which pre-trains a masked language model with significantly less computation resource, we adopt a generator-discriminator framework in the word-level pre-training. The generator aims to replace masked words with plausible alternatives; and the discriminator aims to predict whether each word in the sentence is an original word or a substitution. To tailor this framework for sentiment analysis, we mask two types of words for generation, sentiment

*Corresponding author, chenlin@xmu.edu.cn

†Equal contribution

words and non-sentiment words. We increase the portion of masked sentiment words so that the model focuses more on the sentiment expressions.

For sentence-level pre-training, we design a contrastive learning framework to improve the encoded embeddings by the discriminator. The query for the contrastive learning is constructed by masking sentiment expressions in a sentence. The positive example is the original sentence. The negative examples are selected firstly from in-batch samples and then from cross-batch similar samples using an asynchronously updated approximate nearest neighboring (ANN) index. In this way, the discriminator, which will be used as the encoder for downstream tasks, learns to distinguish different sentiment polarities even if they are superficially similar.

Our main contributions are in three folds: 1). SentiWSP strengthens word-level pre-training via masked sentiment word generation and detection, which is more sample-efficient and benefits various sentiment classification tasks; 2). SentiWSP combines word-level pretraining with sentence-level pre-training, which has been underconsidered in previous studies. SentiWSP adopts contrastive learning in the pre-training, where sentences are progressively contrasted with in-batch and cross-batch hard negatives, so that the model is empowered to encode detailed sentiment information of a sentence; 3). We conduct extensive experiments on sentence-level and aspect-level sentiment classification tasks, and show that SentiWSP achieves new state-of-the-art performance on multiple benchmarking datasets.

2 Related Work

Pre-training and Representation Learning Pre-training models has shown great success across various NLP tasks (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019). Existing studies mostly use a Transformer-based (Vaswani et al., 2017) encoder to capture contextual features, along with masked language model (MLM) and/or next sentence prediction (Devlin et al., 2019) as the pre-training tasks. Yang et al. (2019) propose XLNet which is pre-trained using a generalized autoregressive method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. ELECTRA (Clark et al., 2020) is a generator-discriminator framework, where the generator per-

forms the masked token generation and the discriminator performs replaced token detection pre-training task. It is more efficient than MLM because the discriminator models over all input tokens rather than the masked tokens only. Our work improves ELECTRA’s performance on sentiment analysis tasks, by specifying masked sentiment words at word-level pre-training, and combining sentence-level pre-training.

In addition to the pre-training models that encode token representations, sentence-level and passage-level representation learning have undergone rapid development in recent years. A surge of work demonstrates that contrastive learning is an effective framework for sentence- and passage-level representation learning (Meng et al., 2021; Wei et al., 2021; Gao et al., 2021; Li et al., 2021a). The common idea of contrastive learning is to pull together an anchor and a “positive” sample in the embedding space, and push apart the anchor from “negative” samples. Recently, COCO-LM (Meng et al., 2021) creates positive samples by masking and cropping tokens from sentences. Gao et al. (2021) demonstrate that constructing positive pairs with only standard dropout as minimal data augmentation works surprisingly well on the Natural Language Inference (NLI) task. Karpukhin et al. (2020) investigate the impact of different negative sampling strategies for passage representation learning based on the task of passage retrieval and question answering. ANCE (Xiong et al., 2021) adopts approximate nearest neighbor negative contrastive learning, a learning mechanism that selects hard negatives globally from the entire corpus, using an asynchronously updated Approximate Nearest Neighbor (ANN) index. Inspired by COCO-LM (Meng et al., 2021) and ANCE (Xiong et al., 2021), we construct positive samples by masking a span of words from a sentence, and construct cross-batch hard negative samples to enhance the discriminator at sentence-level pre-training.

Pre-trained Models for Sentiment Analysis In the field of sentiment analysis, BERT-PT (Xu et al., 2019) conducts post-training on the corpora which belong to the same domain of the downstream tasks to benefit aspect-level sentiment classification. SKEP (Tian et al., 2020) constructs three sentiment knowledge prediction objectives in order to learn a unified sentiment representation for multiple sentiment analysis tasks. SENTIX (Zhou et al., 2020) investigates domain-invariant senti-

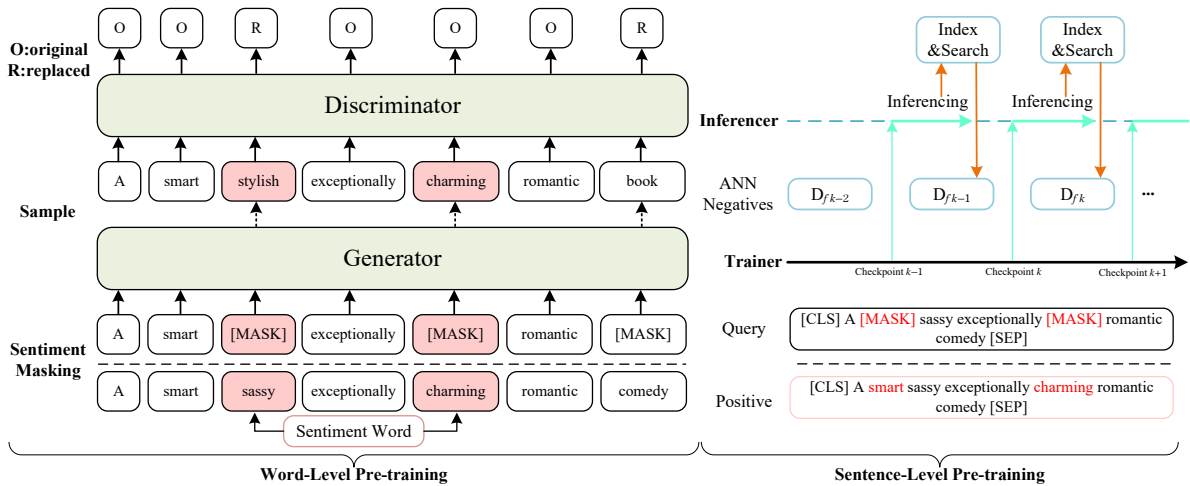


Figure 1: Framework overview of SentiWSP.

ment knowledge from large-scale review datasets, and utilizes it for cross-domain sentiment classification tasks without fine-tuning. SentiBERT (Yin et al., 2020) proposes a two-level attention mechanism on top of the BERT representation to capture phrase-level compositional semantics. SentiLARE (Ke et al., 2020) devises a new pre-training task called label-aware masked language model to construct knowledge-aware language representation. SCAPT (Li et al., 2021b) captures both implicit and explicit sentiment orientation from reviews by aligning the representation of implicit sentiment expressions to those with the same sentiment label.

3 Sentiment-Aware Word-Level and Sentence-Level Pre-training

The overall framework of SentiWSP is depicted in Figure 1. SentiWSP consists of two pre-training phases, namely Word-level pre-training (Sec. 3.1), and Sentence-level pre-training (Sec. 3.2), before fine-tuning (Sec. 3.3) on a downstream sentiment analysis task.

In **word-level pre-training**, an input sentence flows through a word-masking step, followed by a generator to replace the masked words, and a discriminator to detect the replacements. The generator and discriminator are jointly trained in this stage. Then, the training of discriminator continues in **sentence-level pre-training**. Each input sentence is masked at sentiment words to construct a query, while the original sentence is treated as the positive sample. Their embeddings encoded by the discriminator are contrasted to two types of negative samples constructed in an in-batch warm-up

training step and a cross-batch approximate nearest neighbor training step. Finally, the discriminator is **fine-tuned** on the downstream task.

Compared with previous studies, the discriminator in SentiWSP has three advantages. (1) Instead of random token replacement and detection, SentiWSP masks a large portion of sentiment words, and thus the discriminator pays more attention to word-level sentiments. (2) Instead of pure masked token prediction, SentiWSP incorporates context information from all input words via a replacement detection task. (3) SentiWSP combines sentence-level sentiments with word-level sentiments by progressively contrasting a sentence with missing sentiments to a superficially similar sentence.

3.1 Word-Level Pre-training

Word masking Different from previous random word masking (Devlin et al., 2019; Clark et al., 2020), our goal is to corrupt the sentiment of the input sentence.

In detail, we first randomly mask 15% words, the same as ELECTRA (Clark et al., 2020). Then, we use SentiWordNet (Baccianella et al., 2010) to mark the positions of sentiment words in a sentence, and mask the sentiment words until a certain proportion p_w of sentiment words are hidden. We empirically find that the sentiment word masking proportion $p_w = 50\%$ achieves the best results.

In the example in Figure 1 (left), the sentiment words “sassy” and “charming” are masked while “smart” is not masked, “comedy” is masked as a random non-sentiment word.

Generator Next, a generator G processes the masked sentence and generates a corrupted sen-

tence. As in ELECTRA (Clark et al., 2020), the generator is a small sized Transformer. Formally, the sentence is a sequence of words, i.e., $\mathbf{s} = [w_1, w_2, \dots, w_n]$, the mask indicators are denoted as $\mathbf{m} = [m_1, m_2, \dots, m_n]$, $m_t \in \{0, 1\}$, we obtain \mathbf{s}^{mask} from the word masking step. For masked out words, the word is replaced by “MASK”, i.e., $\forall m_t = 1, w_t = [\text{“MASK”}]$. The generator \mathbf{G} encodes the input to contextualized representations $h_{\mathbf{G}}(\mathbf{s}^{\text{mask}}) = [h_1, \dots, h_n]$.

For a given position t , (in our case only positions where $w_t = [\text{“MASK”}]$), the generator \mathbf{G} outputs a probability $p_{\mathbf{G}}(w_t | \mathbf{s}^{\text{mask}})$ for generating a particular token w_t with a softmax layer:

$$p_{\mathbf{G}}(w_t | \mathbf{s}^{\text{mask}}) = \frac{\exp(\mathbf{e}_t^T h_{\mathbf{G}}(\mathbf{s}^{\text{mask}})_t)}{\sum_{w_j} \exp(\mathbf{e}_j^T h_{\mathbf{G}}(\mathbf{s}^{\text{mask}})_t)} \quad (1)$$

where \mathbf{e}_t denotes word embeddings for word w_t .

We then replace the current word w_t with a random sample strategy based on $p_{\mathbf{G}}(w_t | \mathbf{s}^{\text{mask}})$. Sampling introduces randomness and thus it is beneficial for training the discriminator. On the contrary, selecting the word with the highest probability is likely to generate the original word, and the training for discriminator will be more challenging as the discriminator is likely to be trapped to distinguish an original word from a substitution. Formally, the replacing process can be defined as $\forall m_t = 1, \hat{w}_t \sim p_{\mathbf{G}}(w_t | \mathbf{s}^{\text{mask}})$. We denote the corrupted sequence as $\mathbf{s}^{\text{rep}} = [w_1^{\text{rep}}, \dots, w_n^{\text{rep}}]$, where $\forall m_t = 1, w_t^{\text{rep}} = \hat{w}_t$.

Discriminator For the corrupted sentence, the discriminator \mathbf{D} , i.e., a larger sized Transformer, encodes the corrupted sentence to $h_{\mathbf{D}}(\mathbf{s}^{\text{rep}})$, and predicts whether each word w_t comes from the data or the generator, using a sigmoid output layer:

$$\mathbf{D}(\mathbf{s}^{\text{rep}}, t) = \sigma(\mathbf{e}_t^T h_{\mathbf{D}}(\mathbf{s}^{\text{rep}})_t) \quad (2)$$

We jointly train the generator and the discriminator. The generator \mathbf{G} is trained by maximal likelihood estimation, and the discriminator \mathbf{D} is trained by cross entropy.

$$\begin{aligned} \min_{\theta_{\mathbf{G}}, \theta_{\mathbf{D}}} \sum_{\mathbf{s} \in \mathcal{X}} \mathcal{L}_{\mathbf{G}}(\mathbf{s}, \theta_{\mathbf{G}}) + \lambda \mathcal{L}_{\mathbf{D}}(\mathbf{s}, \theta_{\mathbf{D}}) \quad (3) \\ \mathcal{L}_{\mathbf{G}} = \sum_{m_t=1} -\log p_{\mathbf{G}}(w_t | \mathbf{s}^{\text{mask}}) \\ \mathcal{L}_{\mathbf{D}} = \sum_{i=1}^n -\mathbb{I}(w_i^{\text{rep}} = w_i) \log \mathbf{D}(\mathbf{s}^{\text{rep}}, i) \\ - (1 - \mathbb{I}(w_i^{\text{rep}} = w_i)) \log (1 - \mathbf{D}(\mathbf{s}^{\text{rep}}, i)) \end{aligned}$$

where \mathcal{X} denotes a large corpus of raw text and λ is the coefficient of the discriminator loss.

3.2 Sentence-Level Pre-training

For sentence-level pre-training, we follow the contrastive framework in Chen et al. (2020). The goal of contrastive learning is to learn effective representations by pulling together similar samples (i.e., the positive samples) and pushing away different samples (i.e., the negative samples).

One critical question in contrastive learning is how to construct a pair of query (anchor) and positive/negative samples, i.e., (q_i, d_i^+) , (q_i, d_i^-) . As the example shown in Figure 1 (right), given a sentence \mathbf{s}_i from corpus C , we first mask out a certain percentage (70% in this research) of sentiment words in the sentence to construct q_i , and treat the raw sentence as the positive example d_i^+ .

In-batch warm-up training Then we fetch the already trained (in word-level pre-training) discriminator model \mathbf{D} and conduct a warm-up sentence-level training with in-batch negatives. In detail, We feed the input (q_i, d_i^+) to the encoder \mathbf{D} to get the representations \mathbf{f}_i and \mathbf{f}_i^+ , and train the encoder to minimize the distance between the positive pairs within a mini-batch using the neg log-likelihood loss defined as:

$$\min_{\theta_{\mathbf{D}}} - \sum_{i \in \mathcal{B}} \log \frac{\exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_i^+) / \tau)}{\sum_{j=1}^{|\mathcal{B}|} \exp(\text{sim}(\mathbf{f}_i, \mathbf{f}_j^+) / \tau)} \quad (4)$$

where τ is a temperature hyperparameter, $|\mathcal{B}|$ denotes size of the mini-batch \mathcal{B} , $\text{sim}(\cdot, \cdot)$ denotes cosine similarity between two vectors.

Cross-batch approximate nearest neighbor training Since in-batch negatives are unlikely to provide informative samples, we use the Approximate nearest neighbor Negative Contrastive Estimation, (ANCE) (Xiong et al., 2021) to select “hard” negative samples from the entire corpus, to improve the discriminator’s distinguishing power, using an asynchronously updated Approximate Nearest Neighbor (ANN) index. In detail, after the warm-up training of the model \mathbf{D} , we use it to infer the sentence embedding on the entire corpus C and then use ANN search to retrieve top- k negative examples closest to each query. Then we sample t negative examples as hard-negative examples from the top- k negatives. The hyperparameters k and t are set to 100 and 7, respectively.

Model	Sentence-level					Aspect-level			
	IMDB	SST-5	Yelp-2	Yelp-5	MR	Restaurant14	Laptop14		
	Acc	Acc	Acc	Acc	Acc	Acc	MF1	Acc	MF1
BERT (Devlin et al., 2019)	93.87	53.37	97.74	70.16	87.52	83.77	76.06	78.53	73.11
XLNet (Yang et al., 2019)	96.21	56.33	97.41	70.23	89.45	84.93	76.70	80.00	75.88
RoBERTa (Liu et al., 2019)	94.68	54.89	97.98	70.12	89.41	86.07	79.21	81.03	77.16
BERT-PT (Xu et al., 2019)	93.99	53.24	97.77	69.90	87.30	85.86	77.99	78.46	73.82
TransBERT (Li et al., 2019)	94.79	55.56	96.73	69.53	88.69	86.38	78.95	80.06	75.43
SentiBERT (Yin et al., 2020)	94.04	56.87	97.66	69.94	88.59	83.71	75.42	76.87	71.74
SentiLARE (Ke et al., 2020)	95.71	58.59	98.22	71.57	90.82	88.32	81.63	82.16	78.70
SENTIX (Zhou et al., 2020)	94.78	55.57	97.83	\	\	87.32	\	80.56	\
SCAPT (Li et al., 2021b)	\	\	\	\	\	89.11	83.79	82.76	79.15
ELECTRA (Clark et al., 2020)	95.62	57.89	97.87	71.27	90.81	87.32	81.63	82.13	78.25
SentiWSP	96.26	59.32	98.25	71.69	92.41	88.75	82.85	83.69	80.82

Table 1: Overall performance of different models on sentiment classification tasks, “Acc” and “MF1” denote accuracy and macro-F1 score, respectively.

To maintain an up-to-date ANN index two operations are required: (1) inference: refresh the embeddings of all sentences in the corpus with the updated model \mathbf{D} ; and (2) indexing: rebuild the ANN index with the updated embeddings. Although indexing is efficient (Johnson et al., 2021), inferential computing for each batch is expensive as it needs to be passed forward across the entire corpus. In order to balance the time cost between inference and indexing, we use an asynchronous refresh mechanism similar to Guu et al. (2020) and update the ANN index every m steps. As illustrated in the top right part in Figure 1, we construct a trainer to optimize \mathbf{D} , and an inferencer that uses the latest checkpoint (e.g., checkpoint $k - 1$) to recalculate the encoding \mathbf{f}^{k-1} of the entire corpus and update $\text{ANN}_{\mathbf{f}^{k-1}}$. Then, the trainer optimizes a cross-entropy objective function with negative samples generated from $\text{ANN}_{\mathbf{f}^{k-1}}$ and the original positive example pair (q_i, d_i^+) .

$$\min_{\theta_{\mathbf{D}}} \sum_{i \in \mathcal{B}^k} -\log(\text{sim}(\mathbf{f}_i, \mathbf{f}_i^+)) \quad (5)$$

$$- \sum_{j \sim \text{ANN}_{\mathbf{f}^{k-1}}} \log(1 - \text{sim}(\mathbf{f}_i, \mathbf{f}_j))$$

where \mathcal{B}^k is the mini-batch batch at checkpoint k , $\mathbf{f}_i, \mathbf{f}_i^+, \mathbf{f}_j$ indicate the discriminator \mathbf{D} ’s embeddings of the query, positive, and negative samples generated from the asynchronously updated ANN, respectively.

3.3 Fine-tuning

After the pre-training, we fine-tune our model on downstream sentiment analysis tasks. For sentence-

level sentiment classification task, we format the input sequence as: $[CLS], \mathbf{e}_1, \dots, \mathbf{e}_n, [SEP]$, and take the representation at the $[CLS]$ token to predict the sentiment label \mathbf{y} , which indicates the sentiment polarity of the sentence.

For aspect-level sentiment classification task, we format the input sequence as: $[CLS], \mathbf{a}_1, \dots, \mathbf{a}_m, [SEP], \mathbf{e}_1, \dots, \mathbf{e}_n, [SEP]$. where $\mathbf{a}_1, \dots, \mathbf{a}_m$ denotes the phrase of a particular aspect. We fetch the representation at the $[CLS]$ token to predict the sentiment label \mathbf{y} of the sentence in the aspect.

4 Experiment

4.1 Datasets

For SentiWSP pre-training, we use the same English Wikipedia corpus as Devlin et al. (2019). We select 2 million sentences with a maximum length of 128 for the word-level pre-training, and select 500,000 sentences which have 20%-30% proportion of sentiment words for the sentence-level pre-training.

After pre-training, we fine-tune our model on sentence-level sentiment classification benchmarks including Stanford Sentiment Treebank (SST) (Socher et al., 2013), IMDB (Maas et al., 2011), Movie Review (MR) (Pang and Lee, 2005), and Yelp-2/5 (Zhang et al., 2015). For aspect-level sentiment classification tasks, we choose SemEval2014 Task 4 in laptop (Laptop14) and restaurant (Restaurant14) domains (Pontiki et al., 2014). Table 3 shows the statistics of these datasets, including the amount of training, validation, and test

Model	Sentence-level					Aspect-level			
	IMDB	SST-5	Yelp-2	Yelp-5	MR	Restaurant14		Laptop14	
	Acc	Acc	Acc	Acc	Acc	Acc	MF1	Acc	MF1
SentiWSP-base	95.57	58.12	98.08	71.09	90.46	87.14	81.33	82.12	78.34
w/o WP	95.46	57.64	98.02	70.82	90.17	86.82	80.02	81.83	77.87
w/o SP	95.41	57.97	98.01	70.78	90.12	86.75	79.98	81.75	77.54
w/o WP,SP	94.98	57.17	97.59	70.67	89.82	86.13	79.53	81.35	77.23
SentiWSP-large	96.26	59.32	98.25	71.69	92.41	88.75	82.85	83.69	80.82
w/o WP	96.12	58.67	98.21	71.31	91.87	87.87	82.13	82.97	79.35
w/o SP	96.17	58.34	98.17	71.35	91.77	87.56	81.85	82.89	79.13
w/o WP,SP	95.62	57.89	97.87	71.27	90.81	87.32	81.63	82.13	78.25

Table 2: The ablation study results. SP and WP represent sentence-level pre-training and word-level pre-training, respectively. The model without both pre-training is the original ELECTRA model.

Dataset	Train/Valid/Test	Length	#C
SST	8,544 / 1,101 / 2,210	19.2	5
MR	8,534 / 1,078 / 1,050	21.7	2
IMDB	22,500 / 2,500 / 25,000	279.2	2
Yelp-2	504,000 / 56,000 / 38,000	155.3	2
Yelp-5	594,000 / 56,000 / 50,000	156.6	5
Res14	2,163 / 150 / 638	25.6	3
Lap14	3,452 / 150 / 676	30.2	3

Table 3: Statistics of datasets used in our experiments. #C indicates the number of target classes in each dataset.

splits, the average length and the number of classes. Since there is no validation set in MR, IMDB, and yelp-2/5, we randomly select a subset from the training set for validation.

4.2 Baselines

We compare our model with both general purposed pre-trained models and sentiment-aware pre-trained models. For general purposed pre-trained models, we choose BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019) as baselines. For sentiment-aware pre-trained models, we choose BERT-PT (Xu et al., 2019), SentiBERT (Yin et al., 2020), SentiLARE (Ke et al., 2020), SENTIX (Zhou et al., 2020), and SCAPT (Li et al., 2021b). We also implement TransBERT (Li et al., 2019), in the same transfer manner as SentiLARE.*

*We choose review-level sentiment classification on Yelp Dataset Challenge 2019 as the transfer task of TransBERT and the sentiment classification downstream tasks as the target tasks of TransBERT.

4.3 Implementation Details

During pre-training, we use the AdamW optimizer and linear learning rate scheduler, and we set the max sequence length to 128. The learning rate is initialized with $2e-5$ and $1e-5$ for the base and large model, respectively. For word-level pre-training, we use ELECTRA (Clark et al., 2020) initialize **G** and **D**. We set the proportion of sentiment word mask to $p_w = 0.5$ and we keep other hyperparameters the same as ELECTRA. For sentence-level pre-training, we follow the settings of unsupervised SimCSE (Gao et al., 2021) to do the warm-up training, and set the proportion of sentiment word mask $p_s = 0.7$. The detailed batch size and training steps for different level of pre-training are listed in Appendix A.

For fine-tuning, we use the hyperparameters from Clark et al. (2019) for the most parts. We fine-tune 3-5 epochs for sentence-level sentiment classification and 7-10 epochs for aspect-level sentiment classification tasks. The learning rate of the base and large model for the fine-tuning is set to $2e-5$ and $1e-5$, respectively. We use a linear learning rate scheduler with 10% warm-up steps.

4.4 Comparative Results

We list the performance of different models in Table 1. According to the results, we have several findings: (1) SentiWSP consistently outperforms all baselines on sentence-level classification tasks, which demonstrates the superiority of SentiWSP to capture sentence-level semantics. (2) On the aspect-level sentiment classification tasks, the proposed SentiWSP boosts the ACC by 0.93 and increases MF1 by 1.67 on Laptop14 dataset. It also achieves

Model	p_w	IMDB	MR
SentiWSP-only WP	0	95.59	90.83
SentiWSP-only WP	0.3	96.13	91.67
SentiWSP-only WP	0.5	96.17	91.77
SentiWSP-only WP	0.7	95.98	91.07
SentiWSP-only WP	1	95.97	91.26

Table 4: Acc obtained by additionally masking p_w of sentiment words in word-level pre-training, on IMDB and MR.

a competitive performance on Restaurant14 dataset, i.e., the second best among all competitors. (3) SentiWSP is significantly better than ELECTRA, on both sentiment analysis tasks, on all datasets. This observation verifies the effectiveness of the proposed sentiment-aware pretraining strategy.

4.5 Ablation Study

To further investigate the effectiveness of the combining word-level and sentence-level pre-training, we conduct an ablation study with different model sizes (details in Appendix A). From the results (Table 2) we have the following observations: (1) Without sentence-level pre-training and word-level pre-training (i.e., “w/o WP,SP”), SentiWSP degrades to ELECTRA. It performs worst in terms of all metrics. This proves the necessity of tailoring pre-training paradigms for sentiment analysis tasks. (2) The full version of SentiWSP produced the best results across different tasks and datasets. Word-level pre-training and sentence-level pre-training capture sentiment information at different granularity, and combining multi-granularity pre-training is beneficial. (3) Comparing sentence-level pre-training and word-level pre-training, SentiWSP without sentence-level pre-training is generally worse than SentiWSP without word-level pre-training. The performance decline is consistent on Aspect-level sentiment classification task. We think the reason is that the global context is essential for analyzing aspect sentiments, while focusing only on word-level information leads to less robust prediction.

4.6 Impacts of Hyper-parameters

In this section, we explore the impact of different hyper-parameters for the proposed model on IMDB and MR datasets.

Word masking. For word-level pre-training, we experiment with replacing different proportions of sentiment words p_w (Table 4). From the table,

Model	p_s	IMDB	MR
SentiWSP-only SP	0.3	96.02	91.18
SentiWSP-only SP	0.5	95.86	91.36
SentiWSP-only SP	0.7	96.12	91.87
SentiWSP-only SP	1	95.89	91.61

Table 5: Acc obtained by additionally masking p_s of sentiment words in sentence-level pre-training, on IMDB and MR.

Model	k	IMDB	MR
In-batch	N/A	95.85	91.60
ANCE	1	95.97	91.69
ANCE	3	96.15	91.87
ANCE	5	96.12	91.79
ANCE	7	96.26	92.41
ANCE	10	96.27	92.12
ANCE	13	96.19	91.83

Table 6: Acc obtained by using only in-batch negative samples and top- k cross-batch negative samples.

we find that the model performs the worst when $p_w = 0$ (i.e., the same mask strategy as ELECTRA), which verifies our assumption that extra sentiment word masking is beneficial for the model to encode sentiment information. Besides, we find that masking and replacing 50% of the sentiment words yields the best result. We argue the reason is that replacing 50% sentiment words is difficult enough for the model to learn meaningful features, while keeping half of the original sentiment words provides useful clues for the model to detect other sentiment words.

Sentence-level positive sample construction.

Similar to the word-level experiment, we mask different proportions of sentiment words to construct positive samples for sentence-level pre-training. As shown in Table 5, the best performance is achieved by masking 70% of the sentiment words. The underlying reason is that, the ideal positive sample should resemble the query, yet the augmentation can provide additional information for the model to learn meaningful representations. We believe 70% of sentiment word masking is a good balancing point. It is worthy to point out that, even the worst performances, i.e., when $p_s = 0.5$ on IMDB and $p_s = 0.3$ on MR, are better than most of the competitors in Table 1.

Negative sample size. In Table 6, we report

Loss	Similarity	IMDB	MR
Triplet	Dot Product	96.05	91.95
Triplet	Cosine	96.16	92.12
NLL	Dot Product	96.21	92.27
NLL	Cosine	96.26	92.41

Table 7: The impact of loss and similarity functions in sentence-level pre-training.

the results with different negative samples in the sentence-level pre-training. From the table we have two findings: (1) when only in-batch negative samples are used, the model performs worst. We argue the reason is that in-batch negatives are too simple for the model to distinguish from a positive sample, and continue training on these “easy” negatives does not make further improvements. (2) When we increase the cross-batch negative sample size from 1 to 10, the model is provided with more informative negative samples. Therefore, the model can learn more detailed sentiment information and the accuracy is improved. However, when we use a large amount of cross-batch negative samples (e.g., 13), the negative samples vary in quality, and thus the model suffers from less similar negatives.

Similarity and loss function. For sentence-level pre-training, we compare two commonly adopted similarity functions, i.e., cosine distance and dot product, to measure the similarity between two sentence embeddings. The difference between dot product and cosine distance is that dot product does not incorporate L2 normalization. We also compare two widely used loss functions, i.e., Negative Log-Likelihood (NLL) loss and Triplet loss (Schroff et al., 2015), in contrastive learning and ranking problems. The difference between NLL loss and Triplet loss is that Triplet loss compares a positive example and a negative one directly with respect to a query. We report the comparisons in Table 7. From the table we have two observations: (1) With different loss functions, the cosine distance appears to be a more accurate measurement for sentence similarity and outperforms the dot product. (2) The NLL loss produces better results, with different similarity functions, than the Triplet loss.

4.7 Training Loss Convergence

Our final model is trained on 4 NVIDIA Tesla A100 GPUs with a total training time of fewer than 24 hours. For word-level pre-training, we can observe from Figure 3 that the generator and discrimina-

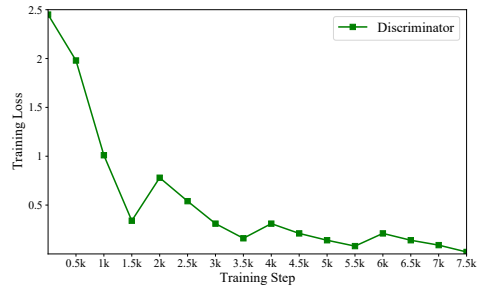


Figure 2: Training loss of sentence-level pre-training.

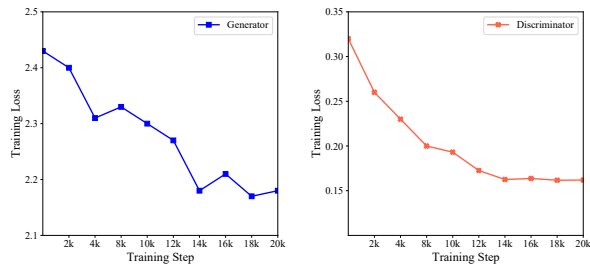


Figure 3: Training loss of the word-level pre-training.

tor compete in joint training and gradually converge within 20,000 steps. For sentence-level pre-training, we can observe from Figure 2 that when the hard-negative example is refreshed every 2000 steps, the loss of the model increases temporarily, which indicates that our ANN search can form a more demanding test for the model and improve the model’s capability on these hard testing cases in the following steps.

5 Conclusion

In this paper, we introduce SentiWSP, which improves pre-training models on sentiment analysis task, by capturing the sentiment information from word-level and sentence-level simultaneously. Extensive experimental results on five sentence-level sentiment classification benchmarks show that SentiWSP establishes new state-of-the-art performance on all of them. We conduct experiment on two aspect-level sentiment classification benchmarks. The results show that SentiWSP beats most existing models on Restaurant14 and achieves new state-of-the-art on Laptop14. We further analyze several hyper-parameters that may affect the model performance, and show that SentiWSP can achieve satisfying performance with respect to different hyper-parameter settings.

Limitations

SentiWSP, as most of the current state-of-the-art pre-training models, requires relatively large

computation resources. As shown in Table 2, SentiWSP-large performs better than SentiWSP-base, the performance divergence is more significant on the MR dataset. We also observe some bad cases when sentiment expressions are implicit suggested in a sentence, i.e., with very few sentiment words, SentiWSP has difficulty in masking and generating, and constructing positive samples. In the future, we plan to devise an adaptive masking mechanism for sentiment words.

Acknowledgements

The project was supported by National Natural Science Foundation of China (No. 61972328, No. 62276219, No. 62036004), Natural Science Foundation of Fujian Province of China (No. 2020J06001), and Youth Innovation Fund of Xiamen (No. 3502Z20206059). We also thank the reviewers for their insightful comments.

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 International Conference on Language Resources and Evaluation*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607.
- Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. Bam! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5931–5937.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR*.
- 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*, pages 4171–4186.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.
- Yaakov HaCohen-Kerner. 2019. Text classification and sentiment analysis in social media for the marketing domain. In *Data Mining in Marketing 15th International Workshop on Data Mining in Marketing, DMM@ICDM 2019, New York, USA, July 19, 2019*, pages 1–10. ibai Publishing.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.
- Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818. Association for Computational Linguistics.
- 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*, pages 6975–6988.
- Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021a. Kfcnet: Knowledge filtering and contrastive learning for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2918–2928. Association for Computational Linguistics.
- Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021b. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 246–256.
- Zhongyang Li, Xiao Ding, and Ting Liu. 2019. Story ending prediction by transferable BERT. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 1800–1806.
- 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](#). *CoRR*, abs/1907.11692.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The*

- 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150.
- Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: correcting and contrasting text sequences for language model pretraining. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23102–23114.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124. The Association for Computer Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 27–35.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Rodrigo Sandoval-Almazán and David Valle-Cruz. 2020. Sentiment analysis of facebook users reacting to political campaign posts. *Digit. Gov. Res. Pract.*, 1(2):12:1–12:13.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society.
- 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.
- Jingyao Tang, Yun Xue, Ziwen Wang, Shaoyang Hu, Tao Gong, Yinong Chen, Haoliang Zhao, and Luwei Xiao. 2022. Bayesian estimation-based sentiment word embedding model for sentiment analysis. *CAAI Trans. Intell. Technol.*, 7(2):144–155.
- Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: sentiment knowledge enhanced pre-training for sentiment analysis. In *ACL*, pages 4067–4076. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations*.
- Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 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*, pages 2324–2335.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 5754–5764.
- Da Yin, Tao Meng, and Kai-Wei Chang. 2020. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706.
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.
- 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.

A Pre-train details

We provide detailed hyper-parameter settings in Table 8 Table 8.

	Parameter	base	large
Word-level pretraining	Batch size	128	64
	Warm up steps	1500	1500
	Max steps	20000	20000
In-batch Warm up	Batch size	64	32
	Warm up steps	500	500
	Max steps	2000	2000
Sentence-level pretraining	Batch size	64	32
	Iteration steps	2000	2000
	Max iterations	4	4
	Max steps	8000	8000

Table 8: SentiWSP pre-training.