

# SKEP: Sentiment Knowledge Enhanced Pre-training for Sentiment Analysis

Hao Tian<sup>†,‡</sup>, Can Gao<sup>†</sup>, Xinyan Xiao<sup>†</sup>, Hao Liu<sup>†</sup>, Bolei He<sup>†</sup>,  
Hua Wu<sup>†</sup>, Haifeng Wang<sup>†</sup>, Feng Wu<sup>‡</sup>

<sup>†</sup>Baidu Inc., Beijing, China    <sup>‡</sup>University of Science and Technology of China  
{tianhao, gaocan01, xiaoxinyan, liuhao24, hebolei, wu\_hua,  
wanghaifeng}@baidu.com, fengwu@ustc.edu.cn

## Abstract

Recently, sentiment analysis has seen remarkable advance with the help of pre-training approaches. However, sentiment knowledge, such as sentiment words and aspect-sentiment pairs, is ignored in the process of pre-training, despite the fact that they are widely used in traditional sentiment analysis approaches. In this paper, we introduce Sentiment Knowledge Enhanced Pre-training (SKEP) in order to learn a unified sentiment representation for multiple sentiment analysis tasks. With the help of automatically-mined knowledge, SKEP conducts sentiment masking and constructs three sentiment knowledge prediction objectives, so as to embed sentiment information at the word, polarity and aspect level into pre-trained sentiment representation. In particular, the prediction of aspect-sentiment pairs is converted into multi-label classification, aiming to capture the dependency between words in a pair. Experiments on three kinds of sentiment tasks show that SKEP significantly outperforms strong pre-training baseline, and achieves new state-of-the-art results on most of the test datasets. We release our code at <https://github.com/baidu/Senta>.

## 1 Introduction

Sentiment analysis refers to the identification of sentiment and opinion contained in the input texts that are often user-generated comments. In practice, sentiment analysis involves a wide range of specific tasks (Liu, 2012), such as sentence-level sentiment classification, aspect-level sentiment classification, opinion extraction and so on. Traditional methods often study these tasks separately and design specific models for each task, based on manually-designed features (Liu, 2012) or deep learning (Zhang et al., 2018).

Recently, pre-training methods (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019;

Yang et al., 2019) have shown their powerfulness in learning general semantic representations, and have remarkably improved most natural language processing (NLP) tasks like sentiment analysis. These methods build unsupervised objectives at word-level, such as masking strategy (Devlin et al., 2019), next-word prediction (Radford et al., 2018) or permutation (Yang et al., 2019). Such word-prediction-based objectives have shown great abilities to capture dependency between words and syntactic structures (Jawahar et al., 2019). However, as the sentiment information of a text is seldom explicitly studied, it is hard to expect such pre-trained general representations to deliver optimal results for sentiment analysis (Tang et al., 2014).

Sentiment analysis differs from other NLP tasks in that it deals mainly with user reviews other than news texts. There are many specific sentiment tasks, and these tasks usually depend on different types of sentiment knowledge including sentiment words, word polarity and aspect-sentiment pairs. The importance of these knowledge has been verified by tasks at different level, for instance, sentence-level sentiment classification (Taboada et al., 2011; Shin et al., 2017; Lei et al., 2018), aspect-level sentiment classification (Vo and Zhang, 2015; Zeng et al., 2019), opinion extraction (Li and Lam, 2017; Gui et al., 2017; Fan et al., 2019) and so on. Therefore, we assume that, by integrating these knowledge into the pre-training process, the learned representation would be more sentiment-specific and appropriate for sentiment analysis.

In order to learn a unified sentiment representation for multiple sentiment analysis tasks, we propose *Sentiment Knowledge Enhanced Pre-training* (SKEP), where sentiment knowledge about words, polarity, and aspect-sentiment pairs are included to guide the process of pre-training. The sentiment knowledge is first automatically mined from unlabeled data (Section 3.1). With the knowledge

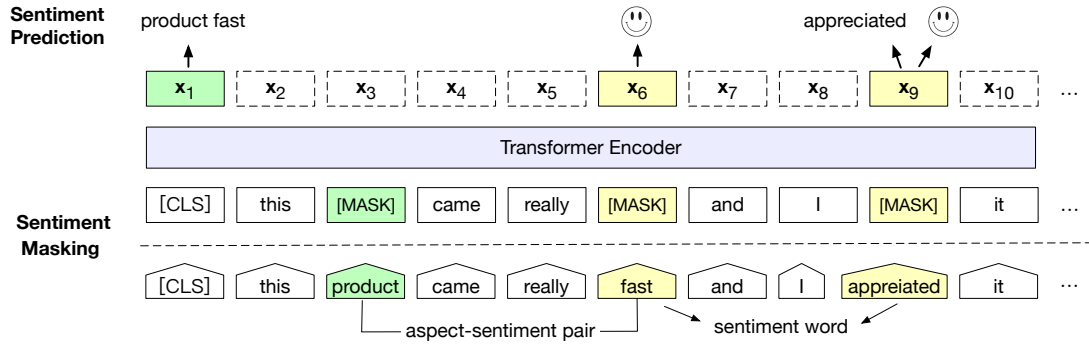


Figure 1: Sentiment Knowledge Enhanced Pre-training (SKEP). SKEP contains two parts: (1) **Sentiment masking** recognizes the sentiment information of an input sequence based on automatically-mined sentiment knowledge, and produces a corrupted version by removing these informations. (2) **Sentiment pre-training objectives** require the transformer to recover the removed information from the corrupted version. The three prediction objectives on top are jointly optimized: Sentiment Word (SW) prediction (on  $x_9$ ), Word Polarity (SP) prediction (on  $x_6$  and  $x_9$ ), Aspect-Sentiment pairs (AP) prediction (on  $x_1$ ). Here, the smiley denotes positive polarity. Notably, on  $x_6$ , only SP is calculated without SW, as its original word has been predicted in the pair prediction on  $x_1$ .

mined, **sentiment masking** (Section 3.2) **removes sentiment information from input texts**. Then, the pre-training model is trained to **recover the sentiment information with three sentiment objectives** (Section 3.3).

SKEP integrates different types of sentiment knowledge together and provides a unified sentiment representation for various sentiment analysis tasks. This is quite different from traditional sentiment analysis approaches, where different types of sentiment knowledge are often studied separately for specific sentiment tasks. To the best of our knowledge, this is the first work that has tackled sentiment-specific representation during pre-training. Overall, our contributions are as follows:

- We propose sentiment knowledge enhanced pre-training for sentiment analysis, which provides a unified sentiment representation for multiple sentiment analysis tasks.
- Three sentiment knowledge prediction objectives are jointly optimized during pre-training so as to embed sentiment words, polarity, aspect-sentiment pairs into the representation. In particular, the pair prediction is converted into multi-label classification to capture the dependency between aspect and sentiment.
- SKEP significantly outperforms the strong pre-training methods RoBERTa (Liu et al., 2019) on three typical sentiment tasks, and achieves new state-of-the-art results on most of the test datasets.

## 2 Background: BERT and RoBERTa

BERT (Devlin et al., 2019) is a self-supervised representation learning approach for pre-training a deep transformer encoder (Vaswani et al., 2017). BERT constructs a self-supervised objective called masked language modeling (MLM) to pre-train the transformer encoder, and relies only on large-size unlabeled data. With the help of pre-trained transformer, downstream tasks have been substantially improved by fine-tuning on task-specific labeled data. We follow the method of BERT to construct masking objectives for pre-training.

BERT learns a transformer encoder that can produce a contextual representation for each token of input sequences. In reality, the first token of an input sequence is a special classification token [CLS]. In fine-tuning step, the final hidden state of [CLS] is often used as the overall semantic representation of the input sequence.

In order to train the transformer encoder, MLM is proposed. Similar to doing a cloze test, MLM predicts the masked token in a sequence from their placeholder. Specifically, parts of input tokens are randomly sampled and substituted. BERT uniformly selects 15% of input tokens. Of these sampled tokens, 80% are replaced with a special masked token [MASK], 10% are replaced with a random token, 10% are left unchanged. After the construction of this noisy version, the MLM aims to predict the original tokens in the masked positions using the corresponding final states.

Most recently, RoBERTa (Liu et al., 2019) significantly outperforms BERT by robust opti-

mization without the change of neural structure, and becomes one of the best pre-training models. RoBERTa also removes the next sentence prediction objective from standard BERT. To verify the effectiveness of our approach, this paper uses RoBERTa as a strong baseline.

### 3 SKEP: Sentiment Knowledge Enhanced Pre-training

We propose SKEP, Sentiment Knowledge Enhanced Pre-training, which incorporates sentiment knowledge by self-supervised training. As shown in Figure 1, SKEP contains **sentiment masking** and **sentiment pre-training objectives**. Sentiment masking (Section 3.2) recognizes the sentiment information of an input sequence based on automatically-mined sentiment knowledge (Section 3.1), and produces a corrupted version by removing this information. Three sentiment pre-training objectives (Section 3.3) require the transformer to recover the sentiment information for the corrupted version.

Formally, sentiment masking constructs a corrupted version  $\tilde{X}$  for an input sequence  $X$  guided by sentiment knowledge  $\mathcal{G}$ .  $x_i$  and  $\tilde{x}_i$  denote the  $i$ -th token of  $X$  and  $\tilde{X}$  respectively. After masking, a parallel data  $(\tilde{X}, X)$  is obtained. Thus, the transformer encoder can be trained with sentiment pre-training objectives that are supervised by recovering sentiment information using the final states of encoder  $\tilde{x}_1, \dots, \tilde{x}_n$ .

#### 3.1 Unsupervised Sentiment Knowledge Mining

SKEP mines the sentiment knowledge from unlabeled data. As sentiment knowledge has been the central subject of extensive research, SKEP finds a way to integrate former technique of knowledge mining with pre-training. This paper uses a simple and effective mining method based on **Pointwise Mutual Information** (PMI) (Turney, 2002).

PMI method depends only on a small number of sentiment seed words and the word polarity  $WP(s)$  of each seed word  $s$  is given. It first builds a collection of candidate word-pairs where each word-pair contains a seed word, and meet with pre-defined part-of-speech patterns as Turney (2002). Then, the co-occurrence of a word-pair is calculated by PMI as follows:

$$PMI(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)} \quad (1)$$

Here,  $p(\cdot)$  denotes **probability estimated by count**. Finally, the polarity of a word is determined by the difference between its PMI scores with all positive seeds and that with all negative seeds.

$$WP(w) = \sum_{WP(s)=+} PMI(w, s) - \sum_{WP(s)=-} PMI(w, s) \quad (2)$$

If  $WP(w)$  of a candidate word  $w$  is larger than 0, then  $w$  is a positive word, otherwise it is negative.

After mining sentiment words, aspect-sentiment pairs are extracted by simple constraints. An aspect-sentiment pair refers to the mention of an aspect and its corresponding sentiment word. Thus, **a sentiment word with its nearest noun will be considered as an aspect-sentiment pair**. The maximum distance between the aspect word and the sentiment word of a pair is empirically limited to no more than 3 tokens.

Consequently, the mined sentiment knowledge  $\mathcal{G}$  contains a collection of sentiment words with their polarity along with a set of aspect-sentiment pairs. Our research focuses for now the necessity of integrating sentiment knowledge in pre-training by virtue of a relatively common mining method. We believe that a more fine-grained method would further improve the quality of knowledge, and this is something we will be exploring in the nearest future.

#### 3.2 Sentiment Masking

Sentiment masking aims to construct a corrupted version for each input sequence where sentiment information is masked. Our sentiment masking is directed by sentiment knowledge, which is quite different from previous random word masking. This process contains sentiment detection and hybrid sentiment masking that are as follows.

**Sentiment Detection with Knowledge** Sentiment detection recognizes both sentiment words and aspect-sentiment pairs by matching input sequences with the mined sentiment knowledge  $\mathcal{G}$ .

1. **Sentiment Word Detection.** The word detection is straightforward. If a word of an input sequence also occurs in the knowledge base  $\mathcal{G}$ , then this word is seen as a sentiment word.
2. **Aspect-Sentiment Pair Detection.** The detection of an aspect-sentiment pair is similar to

its mining described before. A detected sentiment word and its nearby noun word are considered as an aspect-sentiment pair candidate, and the maximum distance of these two words is limited to 3. Thus, if such a candidate is also found in mined knowledge  $\mathcal{G}$ , then it is considered as an aspect-sentiment pair.

**Hybrid Sentiment Masking** Sentiment detection results in three types of tokens for an input sequence: aspect-sentiment pairs, sentiment words and common tokens. The process of masking a sequence runs in following steps:

1. Aspect-sentiment Pair Masking. At most 2 aspect-sentiment pairs are randomly selected to mask. All tokens of a pair are replaced by [MASK] simultaneously. This masking provides a way for capturing the combination of an aspect word and a sentiment word.
2. Sentiment Word Masking. For those unmasked sentiment words, some of them are randomly selected and all the tokens of them are substituted with [MASK] at the same time. The total number of tokens masked in this step is limited to be less than 10%.
3. Common Token Masking. If the number of tokens in step 2 is insufficient, say less than 10%, this would be filled during this step with randomly-selected tokens. Here, random token masking is the same as RoBERTa.<sup>1</sup>

### 3.3 Sentiment Pre-training Objectives

Sentiment masking produces corrupted token sequences  $\tilde{X}$ , where their sentiment information is substituted with masked tokens. Three sentiment objectives are defined to tell the transformer encoder to recover the replaced sentiment information. The three objectives, Sentiment Word (SW) prediction  $L_{sw}$ , Word Polarity (WP) prediction  $L_{wp}$  and Aspect-sentiment Pair (AP) prediction  $L_{ap}$  are jointly optimized. Thus, the overall pre-training objective  $L$  is:

$$L = L_{sw} + L_{wp} + L_{ap} \quad (3)$$

<sup>1</sup>For each sentence, we would always in total mask 10% of its tokens at step 2 and 3. Among these masked tokens, 79.9% are sentiment words (during step 2) and 20.1% are common words (during step 3) in our experiment.

**Sentiment Word Prediction** Sentiment word prediction is to recover the masked tokens of sentiment words using the output vector  $\tilde{x}_i$  from transformer encoder.  $\tilde{x}_i$  is fed into an output softmax layer, which produces a normalized probability vector  $\hat{y}_i$  over the entire vocabulary. In this way, the sentiment word prediction objective  $L_{sw}$  is to maximize the probability of original sentiment word  $x_i$  as follows:

$$\hat{y}_i = \text{softmax}(\tilde{x}_i \mathbf{W} + \mathbf{b}) \quad (4)$$

$$L_{sw} = - \sum_{i=1}^{i=n} m_i \times y_i \log \hat{y}_i \quad (5)$$

Here,  $\mathbf{W}$  and  $\mathbf{b}$  are the parameters of the output layer.  $m_i = 1$  if  $i$ -th position of a sequence is masked sentiment word<sup>2</sup>, otherwise it equals to 0.  $y_i$  is the one-hot representation of the original token  $x_i$ .

Regardless of a certain similarity to MLM of BERT, our sentiment word prediction has a different purpose. Instead of predicting randomly masking tokens, this sentiment objective selects those sentiment words for self-supervision. As sentiment words play a key role in sentiment analysis, the representation learned here is expected to be more suitable for sentiment analysis.

**Word Polarity Prediction** Word polarity is crucial for sentiment analysis. For example, traditional lexicon-based model (Turney, 2002) directly utilizes word polarity to classify the sentiment of texts. To incorporate this knowledge into the encoder, an objective called word polarity prediction  $L_{wp}$  is further introduced.  $L_{wp}$  is similar to  $L_{sw}$ . For each masked sentiment token  $\tilde{x}_i$ ,  $L_{wp}$  calculated its polarity (positive or negative) using final state  $\tilde{x}_i$ . Then the polarity of target corresponds to the polarity of the original sentiment word, which can be found from the mined knowledge.

**Aspect-sentiment Pair Prediction** Aspect sentiment pairs reveal more information than sentiment words do. Therefore, in order to capture the dependency between aspect and sentiment, an aspect-sentiment pair objective is proposed. Especially, words in a pair are *not* mutually exclusive. This is quite different from BERT, which assumes tokens can be independently predicted.

<sup>2</sup>In sentiment masking, we add common tokens to make up for the deficiency of masked tokens of sentiment words.  $L_{sw}$  also calculates these common tokens, while  $L_{wp}$  does not includes them.



We thus conduct aspect-sentiment pair prediction with multi-label classification. We use the final state of classification token [CLS], which denotes representation of the entire sequence, to predict pairs. sigmoid activation function is utilized, which allows multiple tokens to occur in the output at the same time. The aspect-sentiment pair objective  $L_{ap}$  is denoted as follows:

$$\hat{\mathbf{y}}_a = \text{sigmoid}(\tilde{\mathbf{x}}_1 \mathbf{W}_{ap} + \mathbf{b}_{ap}) \quad (6)$$

$$L_{ap} = - \sum_{a=1}^{a=A} \mathbf{y}_a \log \hat{\mathbf{y}}_a \quad (7)$$

Here,  $\mathbf{x}_1$  denotes the output vector of [CLS].  $A$  is the number of masked aspect-sentiment pairs in a corrupted sequence.  $\hat{\mathbf{y}}_a$  is the word probability normalized by sigmoid.  $\mathbf{y}_a$  is the sparse representation of a target aspect-sentiment pair. Each element of  $\mathbf{y}_a$  corresponds to one token of the vocabulary, and equals to 1 if the target aspect-sentiment pair contains the corresponding token.<sup>3</sup> As there are multiple elements of  $\mathbf{y}_a$  equals to 1, the predication here is multi-label classification.<sup>4</sup>

## 4 Fine-tuning for Sentiment Analysis

We verify the effectiveness of SKEP on three typical sentiment analysis tasks: sentence-level sentiment classification, aspect-level sentiment classification, and opinion role labeling. On top of the pre-trained transformer encoder, an output layer is added to perform task-specific prediction. The neural network is then fine-tuned on task-specific labeled data.

**Sentence-level Sentiment Classification** This task is to classify the sentiment polarity of an input sentence. The final state vector of classification token [CLS] is used as the overall representation of an input sentence. On top of the transformer encoder, a classification layer is added to calculate the sentiment probability based on the overall representation.

**Aspect-level Sentiment Classification** This task aims to analyze fine-grained sentiment for an aspect when given a contextual text. Thus, there are two parts in the input: aspect description and

<sup>3</sup>This means that the dimension of  $\mathbf{y}_a$  equals to the vocabulary size of pre-training method, which is 50265 in our experiment.

<sup>4</sup>It is possible to predict masked pairs with CRF-layer. However, it is more than 10-times slower than multi-label classification, thus could not be used in pre-training.

Dataset	Train	Dev	Test
SST-2	67k	872	1821
Amazon-2	3.2m	400k	400k
Sem-R	3608	-	1120
Sem-L	2328	-	638
MPQA2.0	287	100	95

Table 1: Numbers of samples for each dataset. Sem-R and Sem-L refer to restaurant and laptop parts of SemEval 2014 Task 4.

Dataset	Learning Rate	Batch	Epoch
SST-2	1e-5, 2e-5, 3e-5	16, 32	10
Amazon-2	2e-5, 5e-5	4	3
Sem-R	3e-5	16	5
Sem-L	3e-5	16	5
MPQA2.0	3e-5	16	5

Table 2: Hyper-parameters for fine-tuning on each dataset. Batch and Epoch indicate batch size and maximum epoch respectively.

contextual text. These two parts are combined with a separator [SEP], and fed into the transformer encoder. This task also utilizes the final state of the first token [CLS] for classification.

**Opinion Role Labeling** This task is to detect fine-grained opinion, such as holder and target, from input texts. Following SRL4ORL (Marasović and Frank, 2018), this task is converted into sequence labeling, which uses BIOS scheme for labeling, and a CRF-layer is added to predict the labels.<sup>5</sup>

## 5 Experiment

### 5.1 Dataset and Evaluation

A variety of English sentiment analysis datasets are used in this paper. Table 1 summarizes the statistics of the datasets used in the experiments. These datasets contain three types of tasks: (1) For sentence-level sentiment classification, Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) and Amazon-2 (Zhang et al., 2015) are used. In Amazon-2, 400k of the original training data are reserved for development. The performance is evaluated in terms of accuracy. (2) Aspect-level sentiment classification is evaluated on Semantic Eval

<sup>5</sup>All the pretraining models, including our SKEP and baselines use CRF-Layer here, thus their performances are comparable.

Model	Sentence-Level		Aspect-Level		Opinion Role	
	SST-2	Amazon-2	Sem-L	Sem-R	MPQA-Holder	MPQA-Target
Previous SOTA	<b>97.1</b> <sup>1</sup> *	97.37 <sup>2</sup>	81.35 <sup>3</sup>	87.89 <sup>4</sup>	83.67/77.12 <sup>5</sup>	81.59/73.16 <sup>5</sup>
RoBERTa <sub>base</sub>	94.9	96.61	78.11	84.93	81.89/77.34	80.23/72.19
RoBERTa <sub>base</sub> + SKEP	96.7	96.94	81.32	87.92	84.25/79.03	82.77/74.82
RoBERTa <sub>large</sub>	96.5	97.33	79.22	85.88	83.52/78.59	81.74/75.87
RoBERTa <sub>large</sub> + SKEP	97.0	<b>97.56</b>	<b>81.47</b>	<b>88.01</b>	<b>85.77/80.99</b>	<b>83.59/77.41</b>

Table 3: Comparison with RoBERTa and previous SOTA. For MPQA, here reports both binary-F1 and prop-F1 as (Marasović and Frank, 2018), which are split by a slash. The scores of previous SOTA come from: <sup>1</sup>(Raffel et al., 2019; Lan et al., 2019); <sup>2</sup>(Xie et al., 2019); <sup>3</sup>(Zhao et al., 2019); <sup>4</sup>(Rietzler et al., 2019); <sup>5</sup>(Marasović and Frank, 2018). The SOTA score of SST-2 is from GLUE leaderboard (Wang et al., 2018) on December 1, 2019, and the system is based on ensemble-model.

Model	Sentence-Level		Aspect-Level		Opinion Role	
	SST-2 dev	Amazon-2	Sem-L	Sem-R	MPQA-Holder	MPQA-Target
RoBERTa <sub>base</sub>	95.21	96.61	78.11	84.93	81.89/77.34	80.23/72.19
+ Random Token	95.57	96.73	78.89	85.77	82.71/77.71	80.86/73.01
+ SW	96.38	96.82	80.13	86.92	82.95/77.63	81.18/73.15
+ SW + WP	96.51	96.87	80.32	87.25	82.97/77.82	81.09/73.24
+ SW + WP + AP	96.87	96.94	81.32	87.92	84.25/79.03	82.77/74.82
+ SW + WP + AP-I	96.89	96.93	81.19	87.71	84.01/78.36	82.69/74.36

Table 4: Effectiveness of objectives. SW, WP, AP refers to pre-training objectives: Sentiment Word prediction, Word Polarity prediction and Aspect-sentiment Pair prediction. “Random Token” denotes random token masking used in RoBERTa. AP-I denotes predicting words in an Aspect-sentiment Pair Independently.

2014 Task4 (Pontiki et al., 2014). This task contains both restaurant domain and laptop domain, whose accuracy is evaluated separately. (3) For opinion role labeling, MPQA 2.0 dataset (Wiebe et al., 2005; Wilson, 2008) is used. MPQA aims to extract the targets or the holders of the opinions. Here we follow the method of evaluation in SRL4ORL (Marasović and Frank, 2018), which is released and available online. 4-folder cross-validation is performed, and the F-1 scores of both holder and target are reported.

To perform sentiment pre-training of SKEP, the training part of Amazon-2 is used, which is the largest dataset among the list in Table 1. Notably, the pre-training only uses raw texts without any sentiment annotation. To reduce the dependency on manually-constructed knowledge and provide SKEP with the least supervision, we only use 46 sentiment seed words. Please refers to the appendix for more details about seed words.

## 5.2 Experiment Setting

We use RoBERTa (Liu et al., 2019) as our baseline, which is one of the best pre-training mod-

els. Both base and large versions of RoBERTa are used. RoBERTa<sub>base</sub> and RoBERTa<sub>large</sub> contain 12 and 24 transformer layers respectively. As the pre-training method is quite costly in term of GPU resources, most of the experiments are done on RoBERTa<sub>base</sub>, and only the main results report the performance on RoBERTa<sub>large</sub>.

For SKEP, the transformer encoder is first initialized with RoBERTa, then is pre-trained on sentiment unlabeled data. An input sequence is truncated to 512 tokens. Learning rate is kept as  $5e-5$ , and batch-size is 8192. The number of epochs is set to 3. For the fine-tuning of each dataset, we run 3 times with random seeds for each combination of parameters (Table 2), and choose the medium checkpoint for testing according to the performance on the development set.

## 5.3 Main Results

We compare our SKEP method with the strong pre-training baseline RoBERTa and previous SOTA. The result is shown in Table 3.

Comparing with RoBERTa, SKEP significantly and consistently improves the performance on both

From	Model	Sentence Samples	Prediction
SST-2	RoBERTa	<u>altogether</u> , <u>this is successful</u> as a film , <u>while</u> at the same time <u>being a most</u> touching reconsideration of the familiar <u>masterpiece</u> .	positive
	SKEP	<u>altogether</u> , <u>this is successful</u> as a film , <u>while</u> at the same time <u>being a most</u> touching reconsideration of the familiar <u>masterpiece</u> .	positive
Sem-L	RoBERTa	<u>I got this at an amazing price</u> from <u>Amazon</u> and <u>it</u> arrived just in time .	negative
	SKEP	<u>I got this at an amazing price</u> from <u>Amazon</u> and <u>it</u> arrived just in time .	positive

Table 5: Visualization of chosen samples. Words above wavy underline are mean sentiment words, and words above double underlines mean aspects. Color depth denotes importance for classification. The deeper color means more importance. The color depth is calculated by the attention weights with the classification token [CLS].

Model	SST-2 dev	Sem-L	Sem-R
Sent-Vector	96.87	81.32	87.92
Pair-Vector	96.91	81.38	87.95

Table 6: Comparison of vector used for aspect-sentiment pair prediction. Sent-Vector uses sentence representation (output vector of [CLS]) for prediction, while pair-vector uses the concatenation of output vectors of the two words in a pair.

base and large settings. Even on RoBERTa<sub>large</sub>, SKEP achieves an improvement of up to 2.4 points. According to the task types, SKEP achieves larger improvements on fine-grained tasks, aspect-level classification and opinion role labeling, which are supposed to be more difficult than sentence-level classification. We think this owes to the aspect-sentiment knowledge that is more effective for these tasks. Interestingly, “RoBERTa<sub>base</sub> + SKEP” always outperforms RoBERTa<sub>large</sub>, except on Amazon-2. As the large version of RoBERTa is computationally expensive, the base version of SKEP provides an efficient model for application. Compared with previous SOTA, SKEP achieves new state-of-the-art results on almost all datasets, with a less satisfactory result only on SST-2.

Overall, through comparisons of various sentiment tasks, the results strongly verify the necessity of incorporating sentiment knowledge for pre-training methods, and also the effectiveness of our proposed sentiment pre-training method.

## 5.4 Detailed Analysis

**Effect of Sentiment Knowledge** SKEP uses an additional sentiment data for further pre-training and utilizes three objectives to incorporate three types of knowledge. Table 4 compares the contributions of these factors. Further pre-training with random sub-word masking of Amazon, Roberta<sub>base</sub> obtains some improvements. This proves the value

of large-size task-specific unlabeled data. However, the improvement is less evident compared with sentiment word masking. This indicates that the importance of sentiment word knowledge. Further improvements are obtained when word polarity and aspect-sentiment pair objectives are added, confirming the contribution of both types of knowledge. Compare “+SW+WP+AP” with “+Random Token”, the improvements are consistently significant in all evaluated data and is up to about 1.5 points.

Overall, from the comparison of objectives, we conclude that sentiment knowledge is helpful, and more diverse knowledge results in better performance. This also encourages us to use more types of knowledge and use better mining methods in the future.

**Effect of Multi-label Optimization** Multi-label classification is proposed to deal with the dependency in an aspect-sentiment pair. To confirm the necessity of capturing the dependency of words in the aspect-sentiment pair, we also compare it with the method where the token is predicted independently, which is denoted by AP-I. AP-I uses softmax for normalization, and independently predicts each word of a pair as the sentiment word prediction. According to the last line that contains AP-I in Table 4, predicting words of a pair independently do not hurt the performance of sentence-level classification. This is reasonable as the sentence-level task mainly relies on sentiment words. In contrast, in aspect-level classification and opinion role labeling, multi-label classification is efficient and yields improvement of up to 0.6 points. This denotes that multi-label classification does capture better dependency between aspect and sentiment, and also the necessity of dealing with such dependency.

**Comparison of Vector for Aspect-Sentiment Pair Prediction** SKEP utilizes the sentence rep-

resentation, which is the final state of classification token [CLS], for aspect-sentiment pair prediction. We call this Sent-Vector methods. Another way is to use the concatenation of the final vectors of the two words in a pair, which we call Pair-Vector. As shown in Table 6, the performances of these two decisions are very close. We suppose this dues to the robustness of the pre-training approach. As using a single vector for prediction is more efficient, we use final state of token [CLS] in SKEP.

**Attention Visualization** Table 5 shows the attention distribution of final layer for the [CLS] token when we adopt our SKEP model to classify the input sentences. On the SST-2 example, despite RoBERTa gives a correct prediction, its attention about sentiment is inaccurate. On the Sem-L case, RoBERTa fails to attend to the word “amazing”, and produces a wrong prediction. In contrast, SKEP produces correct predictions and appropriate attention of sentiment information in both cases. This indicates that SKEP has better interpretability.

## 6 Related Work

**Sentiment Analysis with Knowledge** Various types of sentiment knowledge, including sentiment words, word polarity, aspect-sentiment pairs, have been proved to be useful for a wide range of sentiment analysis tasks.

Sentiment words with their polarity are widely used for sentiment analysis, including sentence-level sentiment classification (Taboada et al., 2011; Shin et al., 2017; Lei et al., 2018; Barnes et al., 2019), aspect-level sentiment classification (Vo and Zhang, 2015), opinion extraction (Li and Lam, 2017), emotion analysis (Gui et al., 2017; Fan et al., 2019) and so on. Lexicon-based method (Turney, 2002; Taboada et al., 2011) directly utilizes polarity of sentiment words for classification. Traditional feature-based approaches encode sentiment word information in manually-designed features to improve the supervised models (Pang et al., 2008; Agarwal et al., 2011). In contrast, deep learning approaches enhance the embedding representation with the help of sentiment words (Shin et al., 2017), or absorb the sentiment knowledge through linguistic regularization (Qian et al., 2017; Fan et al., 2019).

Aspect-sentiment pair knowledge is also useful for aspect-level classification and opinion extraction. Previous works often provide weak supervision by this type of knowledge, either for aspect-

level classification (Zeng et al., 2019) or for opinion extraction (Yang et al., 2017; Ding et al., 2017).

Although studies of exploiting sentiment knowledge have been made throughout the years, most of them tend to build a specific mechanism for each sentiment analysis task, so different knowledge is adopted to support different tasks. Whereas our method incorporates diverse knowledge in pre-training to provide a unified sentiment representation for sentiment analysis tasks.

**Pre-training Approaches** Pre-training methods have remarkably improved natural language processing, using self-supervised training with large scale unlabeled data. This line of research is dramatically advanced very recently, and various types of methods are proposed, including ELMO (Peters et al., 2018), GPT (Radford et al., 2018), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and so on. Among them, BERT pre-trains a bidirectional transformer by randomly masked word prediction, and have shown strong performance gains. RoBERTa (Liu et al., 2019) further improves BERT by robust optimization, and become one of the best pre-training methods.

Inspired by BERT, some works propose fine-grained objectives beyond random word masking. SpanBERT (Joshi et al., 2019) masks the span of words at the same time. ERNIE (Sun et al., 2019) proposes to mask entity words. On the other hand, pre-training for specific tasks is also studied. GlossBERT (Huang et al., 2019) exploits gloss knowledge to improve word sense disambiguation. SenseBERT (Levine et al., 2019) uses WordNet super-senses to improve word-in-context tasks. A different ERNIE (Zhang et al., 2019) exploits entity knowledge for entity-linking and relation classification.

## 7 Conclusion

In this paper, we propose Sentiment Knowledge Enhanced Pre-training for sentiment analysis. Sentiment masking and three sentiment pre-training objectives are designed to incorporate various types of knowledge for pre-training model. Thought conceptually simple, SKEP is empirically highly effective. SKEP significantly outperforms strong pre-training baseline RoBERTa, and achieves new state-of-the-art on most datasets of three typical specific sentiment analysis tasks. Our work verifies the necessity of utilizing sentiment knowledge for pre-training models, and provides a unified senti-



ment representation for a wide range of sentiment analysis tasks.

In the future, we hope to apply SKEP on more sentiment analysis tasks, to further see the generalization of SKEP, and we are also interested in exploiting more types of sentiment knowledge and more fine-grained sentiment mining methods.

## Acknowledgments

We thanks Qinfei Li for her valuable comments. We also thank the anonymous reviewers for their insightful comments. This work was supported by the National Key Research and Development Project of China (No. 2018AAA0101900).

## References

- Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. [Sentiment analysis of twitter data](#). In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*.
- Jeremy Barnes, Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2019. [Lexicon information in neural sentiment analysis: a multi-task learning approach](#). In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *NAACL 2019*.
- Ying Ding, Jianfei Yu, and Jing Jiang. 2017. [Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction](#). In *AAAI 2017*.
- Chuang Fan, Hongyu Yan, Jiachen Du, Lin Gui, Lidong Bing, Min Yang, Ruifeng Xu, and Ruibin Mao. 2019. [A knowledge regularized hierarchical approach for emotion cause analysis](#). In *EMNLP 2019*.
- Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. [A question answering approach for emotion cause extraction](#). In *EMNLP 2017*.
- Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. [GlossBERT: BERT for word sense disambiguation with gloss knowledge](#). In *EMNLP 2019*.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. [What does BERT learn about the structure of language?](#) In *ACL 2019*.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.
- Zhen-Zhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.
- Zeyang Lei, Yujiu Yang, Min Yang, and Yi Liu. 2018. [A multi-sentiment-resource enhanced attention network for sentiment classification](#). In *ACL 2018*.
- Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. [Sensebert: Driving some sense into bert](#).
- Xin Li and Wai Lam. 2017. [Deep multi-task learning for aspect term extraction with memory interaction](#). In *EMNLP 2017*.
- Bing Liu. 2012. Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies 5.1 (2012): 1-167*.
- 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*.
- Ana Marasović and Anette Frank. 2018. [SRL4ORL: Improving opinion role labeling using multi-task learning with semantic role labeling](#). In *NAACL 2018*.
- Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [SemEval-2014 task 4: Aspect based sentiment analysis](#). In *SemEval 2014*.
- Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. 2017. [Linguistically regularized LSTM for sentiment classification](#). In *ACL 2017*.
- Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#).
- Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. *ArXiv*, abs/1908.11860.

- Bonggun Shin, Timothy Lee, and Jinho D. Choi. 2017. [Lexicon integrated CNN models with attention for sentiment analysis](#). In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *EMNLP 2013*.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. [Ernie: Enhanced representation through knowledge integration](#). *arXiv preprint arXiv:1904.09223*.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberley Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. [Learning sentiment-specific word embedding for twitter sentiment classification](#). In *ACL 2014*.
- Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL 2002*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*.
- Duy-Tin Vo and Yue Zhang. 2015. [Target-dependent twitter sentiment classification with rich automatic features](#). In *IJCAI 2015*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. [Annotating expressions of opinions and emotions in language](#). *Language Resources and Evaluation*.
- Theresa Ann Wilson. 2008. [Fine-grained subjectivity and sentiment analysis: Recognizing the intensity, polarity, and attitudes of private states](#).
- Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. [Unsupervised data augmentation](#). *CoRR*, abs/1904.12848.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#).
- Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *ArXiv*, abs/1703.06345.
- Ziqian Zeng, Wenxuan Zhou, Xin Liu, and Yangqiu Song. 2019. [A variational approach to weakly supervised document-level multi-aspect sentiment classification](#). In *NAACL 2019*.
- Lei Zhang, Shuai Wang, and Bing Liu. 2018. [Deep learning for sentiment analysis : A survey](#). *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *NIPS 2015*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: Enhanced language representation with informative entities](#). In *ACL 2019*.
- Pinlong Zhao, Linlin Hou, and Ou Wu. 2019. [Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification](#). *CoRR*, abs/1906.04501.

## A Appendix

For sentiment knowledge mining, we construct 46 sentiment seed words as follows. We first count the 9,750 items of [Qian et al. \(2017\)](#) on training data of Amazon-2, and get 50 most frequent sentiment words. Then, we manually filter out inappropriate words from these 50 words in a few minutes and finally get 46 sentiment words with polarities (Table 7). The filtered words are *need*, *fun*, *plot* and *fine* respectively, which are all negative words.

positive word	great, good, like, just, will, well, even, love, best, better, back, want, recommend, worth, easy, sound, right, excellent, nice, real, fun, sure, pretty, interesting, stars
negative word	too, little, bad, game, down, long, hard, waste, disappointed, problem, try, poor, less, boring, worst, trying, wrong, least, although, problems, cheap

Table 7: Sentiment seed words used in our experiment.