KESA: A Knowledge Enhanced Approach To Sentiment Analysis

Qinghua Zhao, Shuai Ma, Shuo Ren
SKLSDE Lab, Beihang University, Beijing, China
{zhaoqh, mashuai, shuoren}@buaa.edu.cn

Abstract

Though some recent works focus on injecting sentiment knowledge into pre-trained language models, they usually design mask and reconstruction tasks in the post-training phase. This paper aims to integrate sentiment knowledge in the fine-tuning stage. To achieve this goal, we propose two sentiment-aware auxiliary tasks named sentiment word selection and conditional sentiment prediction and, correspondingly, integrate them into the objective of the downstream task. The first task learns to select the correct sentiment words from the given options. The second task predicts the overall sentiment polarity, with the sentiment polarity of the word given as prior knowledge. In addition, two label combination methods are investigated to unify multiple types of labels in each auxiliary task. Experimental results demonstrate that our approach consistently outperforms baselines (achieving a new state-of-the-art) and is complementary to existing sentiment-enhanced post-trained models. The codes are released at https://github.com/lshowway/KESA.

1 Introduction

Sentence-level sentiment analysis aims to classify the overall sentiment of a sentence, which has received considerable attention in natural language processing (Liu, 2012; Zhang et al., 2018, 2022b). Recently, pre-trained language models (PLMs) have achieved state-of-the-art (SOTA) performance on many natural language processing (NLP) tasks, including sentiment analysis. However, it is still challenging to integrate external knowledge into PLMs (Lei et al., 2018; Xu et al., 2019a; Liu et al., 2020b; Wei et al., 2021; Yang et al., 2021; Cui et al., 2021; Zhang et al., 2022a).

Recently, sentiment dictionary, a commonly used sentiment knowledge, has been injected into PLMs (Wu et al., 2022). A common practice is to post-train (Xu et al., 2019b), i.e., continue pre-training, self-designed tasks on domain-specific corpora. These tasks include sentiment word prediction task, word sentiment prediction task, or aspect-sentiment pairs prediction (Xu et al., 2019a; Tian et al., 2020; Ke et al., 2020; Gururangan et al., 2020; Gu et al., 2020; Tian et al., 2021; Li et al., 2021), just to name a few. Specifically, they are usually designed according to the paradigm of the mask language model (MLM), where sentiment words are first masked and then recovered (including their polarities) in the output layer. Though effective, we argue that these methods can be further boosted by directly injecting sentiment knowledge, e.g., sentiment polarity, into the output layer when fine-tuning the downstream tasks.

In this paper, we aim to inject sentiment knowledge into the fine-tuning phase directly, making it complementary to existing methods. For this aim, we propose two novel auxiliary tasks. The first task is sentiment word selection (SWS), aiming to select the sentiment words that belong to the input from the given options, which comprises of $K + 1$ options (i.e., one ground-truth and K negative words). The second task is conditional sentiment prediction (CSP), which pushes the model to predict the sentence polarity (i.e., sentiment), with the word (within the sentence) polarity given as prior information. It can be seen as a simplified main task (i.e., sentence-level sentiment analysis). Different from existing sentiment polarity prediction task, CSP treats the word sentiment (extracted from the sentiment dictionary) as prior information at the input end instead of as the ground-truth label at the output end. Intuitively, this transformation can reduce the dependency on the quality of the sentiment dictionary. Otherwise, though effective, its interpretability will be impaired. Besides, since more than one type of label (e.g., sentence/word polarity label) is included, two label combination methods, i.e., the joint combination and the conditional combination, are therefore investigated. We are the first (earlier than (Zhang et al., 2022a)) to inject sentiment knowledge in the fine-tuning stage. Our method starts by building the sentiment dictionary out of public resources and recognizing all the sentiment words in the input sentence. Next, each auxiliary task is added to the task-specific (i.e., output) layer. Finally, the auxiliary loss is added to the main loss to achieve the total loss.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 766–776
November 20–23, 2022.

Model	Pre/Post-training 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
	aspect-sentiment pair prediction
SentiLARE	sentiment word prediction
	word polarity prediction
	POS label prediction
	joint prediction
SentiX	sentiment word prediction
	word polarity prediction
	emotion prediction
	rating prediction
KESA	sentiment word selection
	conditional sentiment prediction

Table 1: An overview of tasks. The first block is pre-training tasks, and the second block is knowledge-related tasks. NSP refers to the next sentence prediction.

We conduct experiments to demonstrate the further effectiveness of our proposed approach, and run ablation studies to verify the effectiveness of each auxiliary task. Analysis studies are also performed to compare the impacts of hyper-parameters or modules. With KESA, the performance further outperforms the state-of-the-art by (0.76%, 0.75%) accuracy on MR and SST5, respectively.

2 Related Work

Pre-training Language Models. Pre-trained language models have achieved remarkable improvements in many NLP tasks, and many variants of PLMs have been proposed. For example, GPT, GPT-2 and GPT-3 (Radford et al., 2018, 2019; 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., 2019b). Most PLMs are pre-trained on large-scale unlabeled general corpora by pre-training tasks, pushing models to pay attention to deeper semantic information. Currently, PLMs are the fundamental models across NLP tasks, and the pre-training tasks mentioned above are summarized in Table 1.

Knowledge Enhanced Post-trained Language Models. External knowledge, including linguistic knowledge (e.g., part-of-speech, hyponym and synonym), factual knowledge (including items from Wikidata (Vrandecic, 2012), ConceptNet (Speer et al., 2016) and Wikipedia) or domain-specific knowledge (e.g., sentiment polarity), can boost the generalization abilities of PLMs (Yin et al., 2022). Several works have recently attempted injecting knowledge into PLMs, where the input format or model structure is modified, and knowledge-aware tasks are designed (Zhang et al., 2019; Sun et al., 2021; Liu et al., 2020a; Su et al., 2021; Cui et al., 2021; Yu et al., 2022b,a). For example, ERNIE 3.0 (Sun et al., 2021) appends triples, e.g., (Andersen, Write, Nightingale), ahead of the original input sentence, and designs tasks to predict the relation "Write" in the triple. K-BERT (Liu et al., 2020b) appends triples as branches to each entity (when fine-tuning downstream tasks) involved in the input sentence to form a sentence tree. K-Adapter (Wang et al., 2021) designs adapters and regards them as a plug-in with knowledge representations. These adapters are decoupled from the backbone PLMs and pre-trained from scratch by self-designed tasks, e.g., predicting relations in triples and labels of dependency parser. (Cui et al., 2021) also considers adding two auxiliary training objectives when fine-tuning the dialogue generation task, including conjecturing the meaning of the masked entity and predicting its hypernym. Different from ours, it is also designed according to the paradigm of MLM (i.e., masking entities and predicting their associated attributes in the knowledge base).

Knowledge Enhanced Post-trained Language Models for Sentiment Analysis. Some domain-specific knowledge (including sentiment dictionary) is used for the sentiment analysis task. Generally, these methods inject sentiment-related information into PLMs by designing sentiment-aware tasks and then post-train them on large-scale domain-specific corpora (Tian et al., 2020; Ke et al., 2020; Zhou et al., 2020; Tian et al., 2021). For example, SKEP (Tian et al., 2020) designs sentiment word prediction, word polarity prediction, and aspect-sentiment pair prediction task to enhance PLMs with sentiment words. SentiLARE (Ke et al., 2020) also designs sentiment word prediction, word polarity prediction, and joint prediction tasks. SentiX (Zhou et al., 2020) designs sentiment word prediction, word polarity prediction, emoticon and rating prediction tasks. Table 1 summarizes the tasks mentioned above. Like MLM, they mask sentiment words in the input and then recover their

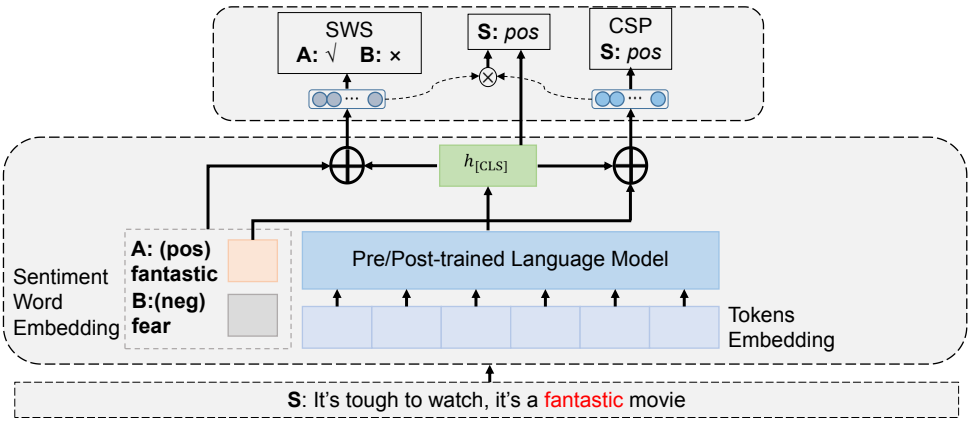


Figure 1: Overview of KESA. Firstly, at the bottom of this figure, the sentence S is tokenized into subwords and input into PLMs to obtain context representation $h_{[\text{CLS}]}$. Meanwhile, sentiment word `fantastic` and its sentiment `positive` are recognized by external sentiment dictionary and a sentiment word `fear` is randomly selected from the sentiment dictionary. Secondly, for the sentiment word selection task, `fantastic` and `fear` are treated as options. For the conditional sentiment prediction task, only the ground-truth sentiment word `fantastic` and its corresponding sentiment `positive` are considered.

related sentiment information in the output. (Tian et al., 2021) associates each aspect term with its corresponding dependency relation types as knowledge to enhance aspect-level sentiment analysis. (Li et al., 2021) enhances aspects and opinions with sentiment knowledge enhanced prompts. Besides, (Zhang et al., 2022a)[1] also injects sentiment knowledge in the fine-tuning phase, it incorporates and updates a lightweight dynamic reweighting adapter when fine-tuning the downstream tasks (we are earlier than this). Our work is different from the above. We propose two novel auxiliary objectives and integrate them with the main objective when fine-tuning the downstream tasks. Furthermore, instead of treating word polarity as a ground-truth label, we treat it as prior knowledge to assist in predicting the overall sentiment. We also investigate two label combination methods to consider several types of labels simultaneously.

3 Methodology

Figure 1 illustrates the framework of KESA. In order to integrate sentiment-related information when fine-tuning the downstream tasks, we propose two straightforward auxiliary tasks. The subsequent subsections will detail the two proposed auxiliary tasks (Section 3.2 and 3.3), two label combination methods (including joint and conditional combination, Section 3.4) and a weighted loss function (Section 3.5) . For convenience, we first give some notations used in the following subsections.

Formally, $L = \{l_1, l_2, \cdots, l_M\}$ denotes the sentiment dictionary with the size of M (i.e., including M sentiment words), and $S = \{w_1, w_2, \cdots, w_N\}$ denotes an input sentence of length N. $P_S \in C$ and $P_w \in Z$ denote the polarity of the sentence S and the word w, respectively, where C is the sentence sentiment polarity label set, and Z is the word sentiment label set. $Y \in \{0, 1\}$ denotes the ascription relationship label set between the word and the sentence, e.g., $Y_{w,S} = 1$ means that the sentiment word w belongs to the sentence S. d is the dimension of embeddings.

3.1 Main Task

The main task, i.e., sentence-level sentiment analysis, is to predict the sentiment label P_S given the input sentence S. Firstly, the input S is passed through PLMs to get the context representation $h_{[\text{CLS}]}$. Then the context representation is fed into a linear layer and a Softmax layer to get the probability $\hat{P}_S$ over sentiment label set, i.e., $\hat{P}_S = \text{Softmax}(W_1 h_{[\text{CLS}]} + b_1)$, where W_1 and b_1 are the model parameters.

3.2 Task A: Sentiment Word Selection

Existing sentiment word prediction tasks usually randomly mask some identified sentiment words in the input, and then predict them in the output layer (in the pre/post-training phase) by computing the probability distribution over the vocabulary of sentiment words. Compared with the number of classes ($|C|$) of the downstream task, the sentiment

[1] We do not take it as a baseline as it is designed for aspect-base sentiment analysis task.

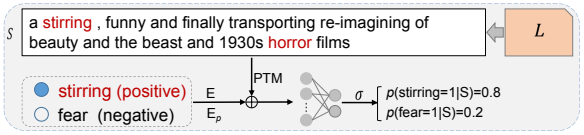


Figure 2: A demonstration of auxiliary task A. The sentence is sampled from SST2 dataset, σ refers to the Softmax layer. It shows that given sentence S, two sentiment word options (i.e., "stirring" and "fear") and their associated sentiment polarities ("positive" and "negative"), "stirring" has more probability of being in S.

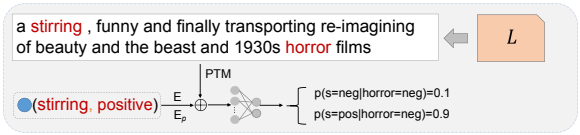


Figure 3: A demonstration of auxiliary task B. This sample shows that the sentiment word, i.e., "horror" and its polarity ("negative") is given as prior knowledge.

word vocabulary size is much larger and directly transferring the above method to the fine-tuning stage may push PLMs to focus on more complex tasks, i.e., the auxiliary tasks. To avoid this issue, we design the sentiment word selection (SWS) task to require PLMs to select the ground-truth sentiment word from given options.

Given a training sample (S, P_S), we first recognize all the sentiment words in S according to the sentiment dictionary L by exact word match. Then, we randomly choose one sentiment word w_i (i.e., positive option) from them and record its sentiment polarity as P_{w_i}. Meanwhile, we randomly sample one sentiment word from L as w_j (i.e., negative option) and record its sentiment polarity as P_{w_j} ($w_j \neq w_i$). Next, we tokenize S into a subwords sequence, add "[CLS]" ahead of the sequence, lookup each subword embedding and input them into PLMs. The first token representation ($h_{[\text{CLS}]}$) of the last layer of PLMs is treated as the context representation (from the view of the representations of sentiment word options).

Meanwhile, we extract the embeddings of the sentiment word options w_i, w_j as e_i and e_j, and the embeddings of its sentiment polarity p_{w_i}, p_{w_j} as e'_i and e'_j, respectively. For each option, we add the context representation, word and its polarity embedding together, and then input them into a linear layer and a Softmax layer to compute the probability $\hat{O}_A = \{\hat{o}_i, \hat{o}_j\}$ over the given options,

$$\hat{o}_x = \text{Softmax}(W_x(h_{[\text{CLS}]} + e_x + e'_x)), x \in \{i, j\} \quad (1)$$

b_x is omitted in Eq. 1, and $W_x, x \in \{i, j\}$ refers to model parameters.

Figure 2 gives an example of the procedure of SWS. In this example, "stirring", "funny", "beauty" and "horror" are first recognized as sentiment words. "stirring" is then randomly selected as the positive option, and "fear" is randomly sampled as a negative option. The sentence S is input into PLMs to get the context representation $h_{[\text{CLS}]}$. Meanwhile, the word embeddings of "stirring" and "fear" are lookup from the sentiment word embedding table $E \in \mathbb{R}^{|V_1| \times d}$, where V_1 refers to sentiment word vocabulary. Correspondingly, their polarity embeddings are looked up from polarity embedding table $E_p \in \mathbb{R}^{|Z| \times d}$. E and E_p can be initialized from scratch and updated during the training, or cached pre-trained embeddings and frozen during the training. Subsequently, $h_{[\text{CLS}]}$ is added to the word and polarity embeddings of the positive (or negative) option, to produce sentiment-enhanced (or polluted) context representation, which is then used to compute the probability of being the ground-truth.

3.3 Task B: Conditional Sentiment Prediction

Existing word polarity prediction tasks usually replace sentiment words with "`[MASK]`" in the input, and recover their sentiment labels in the output layer (in the post-training stage). In this process, sentiment words and their sentiment labels are extracted by sentiment dictionary or statistical methods, which may be inaccurate. Though effective, we argue there are still challenges in interpretability, since it is hard to discriminate which (domain corpus or sentiment-aware tasks) boosts the performance. To avoid the negative impacts of inaccurate polarity of sentiment words, we design the conditional sentiment prediction task, which treats the polarity of sentiment words as prior information instead of the ground-truth label.

More specifically, given a training sample (S, P_S), similar to SWS, we first choose one sentiment word w_i (i.e., positive option detailed in Section 3.2) from all recognized sentiment words in S, meanwhile recording its sentiment polarity P_{w_i}, sentiment word embedding e_i and its polarity embedding e'_i. Next the sentence S is fed into PLMs to get the context representation $h_{[\text{CLS}]}$. Afterwards, we add e_i and e'_i to $h_{[\text{CLS}]}$ to create sentiment-enhanced context representation, then passing them through a linear layer and a Softmax layer to predict

the probability distribution over sentence sentiment label set C, i.e.,

$$\hat{O}_B = \text{Softmax}(W_3(h_{[\text{CLS}]} + e_i + e'_i) + b_3) \quad (2)$$

where W_3, b_3 are model parameters. CSP learns the influence of a word polarity on the polarity of its assigned sentence. In a broader sense, how local information affects global information. Figure 3 gives an example of the auxiliary task B.

3.4 Label Combination

For each auxiliary task, we need to unify all kinds of labels. To be specific, for the SWS task, in addition to the sentence polarity label P_S, we also need to consider the word ascription label Y. Correspondingly, for the CSP task, sentence polarity P_S and word polarity P_w are both involved. Intuitively, multiple kinds of labels can describe the input sentence from different perspectives, and encourage the model to leverage different helpful information simultaneously (Caruana, 1997). To treat the involved label types in a unified manner, we explore two types of combination methods. The first one is joint combination, which models the joint probability distribution of the multiple kinds of labels. This method treats all kinds of labels as a single label defined on the Cartesian product of different labels. The second way is a conditional combination motivated by Lee et al. (2020), which models the conditional probability distribution of multiple kinds of labels, predicting one kind of label with other kinds of labels as prior conditions.

Joint combination. For task A (SWS), given the overall logits $\hat{O}_A$, we need to predict the joint probability distribution of the word ascription label and the sentence polarity label. That is, $p(Y, P_S|\hat{O}_A) \in \mathbb{R}^{|Y|\times|C|}$, where $|Y|$ means the size of label set Y ($\{0, 1\}$) and $|C|$ means the size of label set P_S. For task B (CSP), given the overall logits $\hat{O}_B$ in Eq. 2. We predict the joint distribution of the word polarity label and the sentence polarity label. That is, $p(P_w, P_S|\hat{O}_B) \in \mathbb{R}^{|Z|\times|C|}$, where $|Z|$ means the number of P_w's labels (i.e., {`positive`, `negative`} in our experiment).

Conditional combination. For task A, given the overall logits $\hat{O}_A$, we predict the probability to sentence polarity under the condition that the word ascription label is known, i.e., $p(P_S|\hat{O}_A, Y) \in \mathbb{R}^{|C|}$. To get this, we simply choose the according logits indexed by Y from $\hat{O}_A$ followed by normalization. Similarly, For task B, given the overall logits $\hat{O}_B$ in Eq. 2, the conditional probability of sentence sentiment polarity given the word sentiment polarity is $p(P_S|\hat{O}_B, P_w) \in \mathbb{R}^{|C|}$. For that, we just select the according logits indexed by P_w from $\hat{O}_B$.

3.5 Loss Function

We take cross-entropy loss as our loss function. The loss function is defined as the cross-entropy between the predicted probability (e.g., $\hat{P}_S$, $\hat{O}_A$ and $\hat{O}_B$) and the ground-truth label P_S.

The loss function of the main task is:

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

The loss function of the auxiliary tasks $\mathcal{L}_{aux}$ has the same formulation as Eq. 3, except that the predicted probability is a weighted sum of $\hat{O}_A, \hat{O}_B$:

$$W_4(p(P_S|\hat{O}_A, Y) \,||\, p(P_S|\hat{O}_B, P_w)) \in \mathbb{R}^C \quad (4)$$

where $W_4 \in \mathbb{R}^{2\times1}$ is model parameters, $||$ refers to concatenation, Note that, we omit the bias b_4 in Eq. 4. The final loss is a weighted sum,

$$\mathcal{L} = \mathcal{L}_{main} + \gamma\mathcal{L}_{aux} \quad (5)$$

where γ is loss balance weight and $\gamma \in (0.0, 1.0)$. Notably, the weight of $\mathcal{L}_{main}$ is set to 1.0. We set $\gamma > 0.0$ to ensure that the parameters of the auxiliary tasks can be optimized by backpropagation, and set $\gamma < 1.0$ to prevent the final loss is dominated by the auxiliary task loss and diminishing the performance of the main task (Liu et al., 2019a).

4 Experimental Setup

4.1 Datasets

Four commonly used public sentence-level sentiment analysis datasets are used for the experiment, as shown in Table 2. The datasets include Movie Review (MR) (Pang and Lee, 2005), Stanford Sentiment Treebank (SST2 and SST5) (Socher et al., 2013) and IMDB. For MR and IMDB, we adopt the data split in SentiLARE (Ke et al., 2020), due to the lack of test data in the original dataset. We evaluate the model performance in terms of accuracy.

4.2 Comparison Methods

To demonstrate the further effectiveness of the proposed method, we test the proposed auxiliary tasks on two types of competitive baselines, including

Dataset	#Train/Valid/Test	#W	#C
MR	8,534/1,078/1,050	22	2
SST2	6,920/872/1,821	20	2
SST5	8,544/1,101/2,210	20	5
IMDB	22,500/2,500/25,000	280	2

Table 2: Datasets statistics. The columns are the amount of training/validation/test sets, the average sentence length, and the number of classes, respectively.

popular vanilla pre-trained models (PLMs) and sentiment knowledge enhanced post-trained models.

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

Sentiment Knowledge Enhanced Post-trained Language Models. We also use two methods focusing on leveraging sentiment knowledge as baselines, i.e., SentiLARE (Ke et al., 2020) and SentiX (Zhou et al., 2020). They introduce sentiment knowledge in the pre-training stage by designing sentiment-related tasks (including sentiment word prediction and word polarity prediction task). They continue pre-training vanilla PLMs on million scale domain-specific corpora, i.e., Yelp Dataset Challenge 2019 (6.6 million) for SentiLARE, Yelp Dataset Challenge 2019 and Amazon review dataset (240 million in total) for SentiX. In terms of PLMs, SentiLARE is post-trained on RoBERTa-base version while SentiX is post-trained on BERT-base version.

KESA (Ours). We also utilize the external sentiment knowledge to enhance PLMs when fine-tuning the downstream tasks by designing two auxiliary tasks (i.e., SWS and CSP). KESA is a complementary method to existing models (including vanilla and knowledge-enhanced PLMs).

4.3 Sentiment Dictionary

To build sentiment dictionary, we extract word sentiment (i.e., polarity) from SentiWordNet 3.0 (Baccianella et al., 2010). Since each word in SentiWordNet 3.0 has several usage frequency levels and is linked with different semantic and sentiment scores, we set the sentiment polarity of a word according to its most vital scores (i.e., positive or negative sentiment scores). Take "thirsty" for example, the polarity of the most common usage is "positive" (with a score of 0.25), while the polarity of the third common usage is "negative" (with a score of -0.375). We, therefore, set the polarity of "thirsty" to "negative", considering it has a larger weight of "negative". We adopt this strategy considering a lower sentiment score often means less likely to be a sentiment word.

Model	MR	SST2	SST5	IMDB
BERT*	86.62	91.38	53.52	93.45
XLNet*	88.83	92.75	54.95	94.99
RoBERTa*	89.84	94.00	57.09	95.13
SentiX$^\#$	–	93.30	55.57	94.78
SentiX*	86.81	92.23	55.59	94.62
SentiLARE$^\#$	90.82	–	58.59	95.71
SentiLARE*	90.50	94.58	58.54	95.73
KESA	**91.26**‡	**94.98**‡	**59.26****	**95.83****

Table 3: Overall accuracy (joint combination is adopted here). The marker $\#$ denotes the original reported results while $-$ means not available. The marker $*$ refers to our re-implementation. $**$ and $\ddagger$ indicate that our model significantly outperforms the best baselines with t-test, p-value < 0.01 and 0.05, respectively.

4.4 Implementation Details

We implement our model using HuggingFace's Transformers. The batch size is set to 16 and 32 for IMDB and other datasets, respectively. The learning rate is set to 2e-5 for XLNet, RoBERTa and SentiLARE, and 5e-5 for BERT and SentiX. The input and output formats are consistent with each corresponding PLM. In the meantime, the input sequence length is set to 50, 512, and 128 for MR, IMDB, and other datasets, respectively, to ensure that more than 90% of the samples are covered. Other hyper-parameters are kept by default. We fine-tune each model for three epochs, and the best checkpoints on the development set are used for inference. As for each dataset, we run four times with different random seeds with a reproducible implementation, and the average results are reported. Moreover, to make a fair comparison, all methods use the same seeds for the same dataset. To explore the influence of auxiliary tasks on the main task, we search the loss balance weight γ from $\{0.01, 0.1, 0.5, 1.0\}$. The source code will be released when the paper is accepted.

5 Experimental Results

In this section, we will detail the overall results, and the analysis of loss balance weight, label combination and introduced extra parameters.

Model	MR	SST2	SST5	IMDB
XLNet*	88.83	92.75	54.95	94.99
Δ+SWS	0.22	0.72	0.56	0.04
Δ+CSP	0.48	0.04	0.50	-0.02
Δ+KESA	0.27	0.26	0.99	0.01
BERT*	86.62	91.38	53.52	93.45
Δ+SWS	-0.32	0.08	0.69	0.14
Δ+CSP	-0.17	0.32	0.86	0.06
Δ+KESA	-0.33	0.18	0.61	0.06
SentiX*	86.81	92.23	55.59	94.62
ΔSentiX*	0.19	0.85	2.07	1.17
Δ+SWS	0.50	-0.03	0.15	0.09
Δ+CSP	0.54	0.01	0.24	-0.01
Δ+KESA	0.55	0.29	0.19	-0.05
RoBERTa*	89.84	94.00	57.09	95.13
Δ+SWS	-0.03	0.22	0.13	0.27
Δ+CSP	0.02	0.17	0.15	0.31
Δ+KESA	0.23	0.40	0.09	0.33
SentiLARE*	90.50	94.58	58.54	95.73
ΔSentiLARE*	0.66	0.58	1.45	0.60
Δ+SWS	0.24	0.14	0.75	0.07
Δ+CSP	0.60	0.33	0.05	0.07
Δ+KESA	0.76	0.40	0.72	0.10

Table 4: Ablation studies of each task. "+SWS" and "+CSP" refer to that we fine-tune the models with SWS and CSP solely, respectively. "+KESA" represents that both auxiliary tasks are adopted. The marker $*$ refers to our re-implementation.

5.1 Overall Results

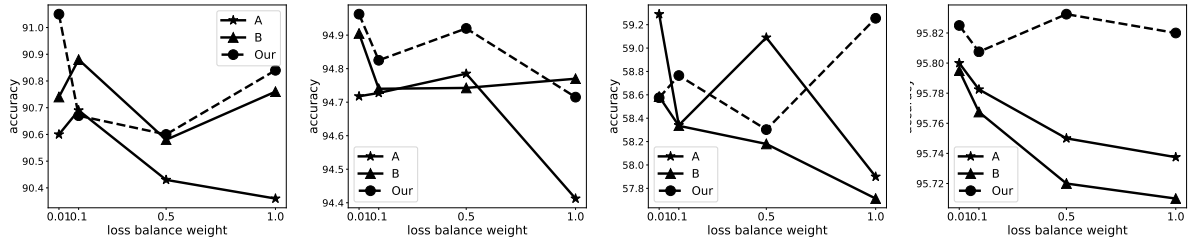
Table 3 reports the results w.r.t. the accuracy. Note that, we only report the results of KESA fine-tuned on the checkpoints released by SentiLARE, since it performs best (others will be detailed next section). We find that through post-training on 240 million samples, SentiX (based on BERT-base) shows improvements of (0.19%, 0.85%, 2.07%, 1.17%) accuracy, respectively. Similarly, post-training on 6.6 million samples, SentiLARE (RoBERTa-base) outperforms the comparad method by (0.66%, 0.58%, 1.45%, 0.60%), respectively. Based on these improvements, KESA can further improve the accuracy by (0.76%, 0.40%, 0.75%, 0.10%), demonstrating that KESA is complementary to existing sentiment-enhanced post-trained methods.

5.2 Ablation Results

To demonstrate the individual benefits of the two auxiliary tasks to each baseline PLMs, we perform ablation experiments and tabulate the results in Table 4. Overall, KESA achieves consistent improvements over both vanilla and sentiment-enhanced PLMs. Adding SWS to the baseline PLMs improves accuracy by a maximum of 0.75%, and further pushes the overall accuracy to 59.29% (SST5), exceeding the previous sentiment-enhanced best of 58.54%. The results verify that the word ascription label pushes the model to focus more on the interactions between the sentiments of word and sentence, and this kind of interactions between sentence sentiment (can be seen as global information) and word sentiment (treated as local information) can promote the main task. With the addition of CSP, the test set accuracy jumped 0.86% from 53.52% to 54.38% (SST5), even improving over the previous best sentiment-enhanced result by 0.60% (MR). The results demonstrate that explicitly adding the sentiment of a word brings more information and lowers the difficulty of the main task. Besides, we can see that integrating KESA with sentiment-enhanced PLMs obtains more gains than that with vanilla PLMs, we attribute this to that the former can achieve better semantic representation of sentiment words. Furthermore, combining the two auxiliary tasks is not necessarily superior to sole use. It is presumably because multiple tasks may promote or compete with each other (negative learning) (Bingel and Søgaard, 2017). Above all, these results remind us that the combinations of multiple tasks need to be carefully analyzed. Even so, KESA still gets further improvements on all evaluated datasets in most cases.

5.3 Analysis on Loss Balance Weight

There are many alternatives to Equation 5 for combining the losses. Previous work on multiple losses used only the sum (Ke et al., 2020). The choice of the loss balance weight γ is also important, as large values such as $\gamma = 1.0$ effectively reduce the weighting function to a simple sum over the losses, while smaller values (e.g., γ = 0.01) allow the loss weights to vary. Therefore, we search the loss balance weight γ from $\{0.01, 0.1, 0.5, 1.0\}$ considering the following detailed considerations. First, we argue that higher auxiliary task weights may dominate the total loss, while smaller weights should be better, and 0.01 is selected. Second, the weights in $(0.0, 1.0]$ should be tested evenly. Figure 4 compares these alternatives, including auxiliary task SWS and CSP, and KESA. It can be observed that, lower loss balance weight generally achieves better performance across most cases. Taking IMDB as an example, as there are more training samples and longer sequence length (512), making it less sen-



(a) MR (b) SST2 (c) SST5 (d) IMDB

Figure 4: Impacts of loss balance weights, from left to right, are the results of MR, SST2, SST5 and IMDB, respectively. A and B refer that auxiliary tasks A and B are tested solely. "Our" refers to KESA.

Model	MR	SST2	IMDB	SST5
SentiX$_{\text{A+JC}}$	87.31	92.20	94.70	55.74
SentiX$_{\text{A+CC}}$	**87.35**	**92.26**	**94.71**	**55.81**
SentiX$_{\text{B+JC}}$	87.35	92.24	94.59	**55.83**
SentiX$_{\text{B+CC}}$	**87.38**	**92.59**	**94.61**	55.74
SentiLARE$_{\text{A+JC}}$	90.69	94.72	95.80	**59.29**
SentiLARE$_{\text{A+CC}}$	**90.74**	**94.91**	**95.83**	59.21
SentiLARE$_{\text{B+JC}}$	90.88	94.91	95.80	58.59
SentiLARE$_{\text{B+CC}}$	**91.10**	**94.99**	**95.84**	**58.97**

Table 5: Comparison of joint combination (JC) and conditional combination (CC) in task A and B.

sitive to seeds, with the decrease of loss balance weight, the advantages gradually increase, indicating that the weight of auxiliary tasks should be a small value to avoid undue impacts on the main task. Although for MR, a dataset with a smaller training set, the results are sensitive to γ, a small γ is also preferred in most cases.

5.4 Analysis on Label Combination

In addition to the auxiliary tasks, KESA also contains a label combination method unifying two different categories of labels (e.g., word/sentence sentiment label). To analyze the relative contribution of the conditional combination method compared to the joint combination method, we run additional comparison experiments that replace the joint combination with just the conditional combination method. Table 5 summarizes the results for all evaluated datasets (SentiX and SentiLARE are selected, as they perform better). Replacing the joint combination with the conditional combination gives a slight improvement for datasets MR, SST2 and IMDB. For dataset SST5, the conditional combination is better than joint combination in some cases (e.g., from 58.59 accuracy to 58.97 for SST5 on the auxiliary task B). Overall the improvements are small compared to the full KESA model. Joint combination is adopted by default in our experiments, as it is slightly easier to implement.

5.5 Introduced Parameters

For SWS, the number of increased parameters is $W_{\{i,j\}} \in \mathbb{R}^{|Y|d\times|C||Y|}, b_{\{i,j\}} \in \mathbb{R}^{|C||Y|}$ (Section 3.2), sentiment word embedding table $E \in \mathbb{R}^{|V_1|\times d}$ and polarity embedding table $E_p \in \mathbb{R}^{|Z|\times d}$. For CSP, the number of extra parameters is $W_3 \in \mathbb{R}^{d\times|Z||C|}, b_3 \in \mathbb{R}^{|Z||C|}$, sentiment word embedding table $E \in \mathbb{R}^{|V_1|\times d}$ and polarity embedding table $E_p \in \mathbb{R}^{|Z|\times d}$. The number of increased parameters induced by combining the two tasks is $W_4 \in \mathbb{R}^{2\times 1}, b_4 \in \mathbb{R}$. Therefore, the total number of parameters induced by KESA is $W_i, W_j, W_3, W_4, b_i, b_j, b_3, b_4$ and E, E_p, where E, E_p is optional since it can be cached (just like GloVe (Pennington et al., 2014)) and kept frozen to avoid introducing much parameters when the sentiment word vocabulary is large. In our experiments, $|C| \leq 5, |Y| = |Z| = 2, d = 768, V_1 = 25,158$.

6 Conclusion

In this paper, we directly integrate sentiment knowledge into the fine-tuning phase. We design two sentiment-aware auxiliary tasks, SWS and CSP. SWS needs to select the correct sentiment words from the given options, while CSP predicts the overall sentiment with the word sentiment given as prior knowledge. Further, we propose joint and conditional label combination methods to unify considered multiple kinds of labels into a single label. Though straightforward and conceptually simple, experiments demonstrate that KESA still further improves over solid baselines, verifying that KESA is complementary to existing sentiment-enhanced PLMs.

7 Acknowledgement

This work is supported in part by NSFC (61925203).

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)*.

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 Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Leyang Cui, Yu Wu, Shujie Liu, and Yue Zhang. 2021. Knowledge enhanced fine-tuning for better handling unseen entities in dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2328–2337.

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.

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.

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.

Zeyang Lei, Yujiu Yang, Min Yang, and Yi Liu. 2018. A multi-sentiment-resource enhanced attention network for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 758–763.

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.

Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang, et al. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *arXiv preprint arXiv:2109.08306*.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020a. Kred: Knowledge-aware document representation for news recommendations. In *Fourteenth ACM Conference on Recommender Systems*, RecSys '20, page 200–209, New York, NY, USA. Association for Computing Machinery.

Shengchao Liu, Yingyu Liang, and Anthony Gitter. 2019a. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9977–9978.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020b. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. 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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *national conference on artificial intelligence*.

Chang Su, Kechun Wu, and Yijiang Chen. 2021. Enhanced metaphor detection via incorporation of external knowledge based on linguistic theories. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1280–1287, Online. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

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.

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.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021. Enhancing aspect-level sentiment analysis with word dependencies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3726–3739, Online. Association for Computational Linguistics.

Denny Vrandecic. 2012. Wikidata: a new platform for collaborative data collection. *the web conference*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiaokai Wei, Shen Wang, Dejiao Zhang, Parminder Bhatia, and Andrew Arnold. 2021. Knowledge enhanced pretrained language models: A compreshensive survey. *arXiv preprint arXiv:2110.08455*.

Yang Wu, Yanyan Zhao, Hao Yang, Song Chen, Bing Qin, Xiaohuan Cao, and Wenting Zhao. 2022. Sentiment word aware multimodal refinement for multimodal sentiment analysis with ASR errors. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1397–1406, Dublin, Ireland. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019a. 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.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019b. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. 2021. A survey of knowledge enhanced pre-trained models. *arXiv preprint arXiv:2110.00269*.

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.

Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. 2022. A survey of knowledge-intensive nlp with pre-trained language models. *arXiv preprint arXiv:2202.08772*.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2022a. Jaket: Joint pre-training of knowledge graph and language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11630–11638.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022b. Dict-bert: Enhancing language

model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918.

Kai Zhang, Kun Zhang, Mengdi Zhang, Hongke Zhao, Qi Liu, Wei Wu, and Enhong Chen. 2022a. Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3599–3610, Dublin, Ireland. Association for Computational Linguistics.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Yiming Zhang, Min Zhang, Sai Wu, and Junbo Zhao. 2022b. Towards unifying the label space for aspect- and sentence-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 20–30, Dublin, Ireland. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

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.