KESA: A Knowledge Enhanced Approach For Sentiment Analysis

Anonymous ACL submission

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. In this paper, we aim to benefit from sentiment knowledge in a lighter way. To achieve this goal, we study sentence-level sentiment analysis and, correspondingly, propose two sentiment-aware auxiliary tasks named sentiment word cloze and conditional sentiment prediction. The first task learns to select the correct sentiment words within the input, given the overall sentiment polarity as prior knowledge. On the contrary, the second task predicts the overall sentiment polarity given the sentiment polarity of the word as prior knowledge. In addition, two kinds of label combination methods are investigated to unify multiple types of labels in each task. We argue that more information can promote the models to learn more profound semantic representation. We implement it in a straightforward way to verify this hypothesis. The experimental results demonstrate that our approach consistently outperforms pre-trained models and is additive to existing knowledge-enhanced post-trained models.

1 Introduction

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

For sentiment analysis task, sentiment lexicon, a kind of commonly used knowledge, has been injected into PTMs. A common practice is to post-train self-designed tasks on domain-specific corpora, e.g., sentiment word prediction task, word sentiment prediction task, aspect-sentiment pairs prediction task or part-of-speech (POS) tag prediction task, and so forth (Xu et al., 2019; Tian et al., 2020; Ke et al., 2020; Gururangan et al., 2020; Gu et al., 2020; Tian et al., 2021; Li et al., 2021). Specifically, they are usually designed according to the paradigm of the mask language model (MLM), where sentiment words are masked and recovered in the input and output layer, respectively. In addition, word sentiment or POS label may be predicted simultaneously. We argue, however, that these methods have the following shortcomings. First, it is computation costly to recover the masked words, since the probability distribution is calculated over the entire vocabulary. Second, it has a greater dependence on the quality of the sentiment lexicon, because sentiment label of words are treated as the ground-truth. This requires the label to be precise, otherwise performance of the tasks and the interpretability of the models will be impaired. Third, extensive domain-specific corpora are used to post-train the proposed tasks. Fourth, sentiment information may lose, because the sentiment words are replaced with "MASK", which can change the semantics of the sentiment of the input.

In this paper, to alleviate the above issues, we propose two novel auxiliary tasks and integrate them into the fine-tuning phase. The first task is sentiment word cloze (SWC), which selects the sentiment words that belong to the input from the options. It contains $K + 1$ options (1 ground-truth word with K negative words), which is much smaller than the vocabulary size of PTMs. The number of calculations and parameters is therefore reduced. The second task is conditional sentiment prediction (CSP), which predicts the sentiment polarity of a sentence, considering the sentiment polarity of the word within it. Conversely, the word sentiment extracted from the sentiment lexicon is treated as prior information at the input end instead of as the ground-truth label at the output end.

Intuitively, this transformation can reduce the dependence on the accuracy of the sentiment lexicon. Also, both auxiliary tasks are injected into the fine-tuning phase, and only task-specific data are used. Note that, the tasks are integrated in the training phase, not the inference phase, to avoid increasing the inference time. Additionally, we do not substitute the selected sentiment words with "MASK" identifiers to prevent loss of critical information. More precisely, our method starts by building the sentiment lexicon out of public resources and recognizing all the sentiment words in the input sentence. Next, two auxiliary tasks are added to the task-specific (output) layer. Additionally, there are also two ways of unifying different types of labels, i.e., the joint combination and the conditional combination, are investigated. Lastly, the auxiliary loss is added to the main loss to achieve the total loss.

Our contributions are outlined below.

- We integrate the sentiment lexicon into the fine-tuning phase by designing two auxiliary tasks. The tasks avoid using a large number of classification classes and reduce dependence on the accuracy of the sentiment lexicon.

- We also investigate the joint and conditional probability combination to unify different types of labels within each task.

- We carry out experiments to demonstrate the effectiveness of our proposed approach. Ablation studies are also performed to verify the effectiveness of each module. The overall improvements on (MR, SST2, SST5, IMDB) are (0.76%, 0.38%, 0.72%, 0.1%), respectively.

2 Related Work

Pre-training Language Models. Pre-trained language models have achieved remarkable improvements in many NLP tasks, and many variants of PTMs 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 PTMs are pre-trained on large-scale unlabeled general corpora by pre-training tasks, which push models to pay attention to deeper semantic information. The pre-training tasks mentioned above are summarized in the first block in Table 1.

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 cloze
	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 next sentence prediction task.

Knowledge Enhanced Post-trained Language Models. Recently, several works have attempted to inject knowledge into pre-trained language models, where input format or model structure is modified, and knowledge-aware tasks are designed (Zhang et al., 2019; Liu et al., 2020b; Sun et al., 2021; Wang et al., 2021; Liu et al., 2020a; Su et al., 2021). 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 involved in the input sentence to form a sentence tree. Hard and soft position encoding is designed to maintain the tree structure. 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 PTMs and pre-trained from scratch by self-designed tasks, e.g., predicting relations in triples and labels of dependency parser.

Knowledge Enhanced Post-trained Language Models for Sentiment Analysis. Sentiment lexicon is usually injected into PTMs by designing sentiment-aware tasks and then post-training on domain-specific corpora (Tian et al., 2020; Ke et al., 2020; Zhou et al., 2020; Tian et al., 2021; Li et al.,
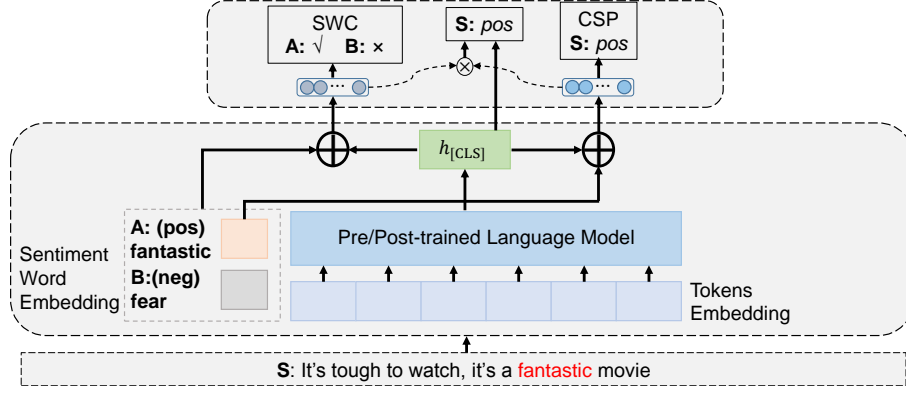
2

Figure 1: Overview of KESA. Firstly, at the bottom of this figure, the sentence **S** is tokenized into subwords and input into PTMs to obtain context state $h_{[CLS]}$. Meanwhile, sentiment word `fantastic` and its sentiment `positive` are recognized by external sentiment lexicon and a sentiment word `fear` is randomly selected from the sentiment lexicon. Secondly, for the Sentiment Word Cloze task, `fantastic` and `fear` are treated as candidates. Their sentiment polarities are included at the same time. For the Conditional Sentiment Prediction task, only the ground-truth sentiment word `fantastic` and its corresponding sentiment are included. Thirdly, the context state, word embedding, and sentiment embedding are concatenated to compute each class's probability (logits). Afterward, the logits (blue circles) are sampled and weighted summed to produce the final probability to sentence-level sentiment. Note that, the context state is also solely used to predict sentence-level sentiment for the main task.

2021). For example, SKEP (Tian et al., 2020) designs sentiment word prediction, word polarity prediction, and aspect-sentiment pair prediction task to enhance PTMs with sentiment knowledge. SentiLARE (Ke et al., 2020) designs sentiment word prediction, word polarity prediction, and word part-of-speech (POS) tag prediction and joint prediction tasks. SentiX (Zhou et al., 2020) designs sentiment word prediction, word polarity prediction, emoticon and rating prediction tasks. Table 1 summarizes the tasks mentioned above. Like MLM, they mask sentiment words in the input and then recover their related information in the output. Besides, for aspect-level sentiment analysis (Tian et al., 2021) associates each aspect term with its corresponding dependency relation types as knowledge. (Li et al., 2021) enhances aspects and opinions with sentiment knowledge enhanced prompts. Our work is different from the above. Firstly, like the word cloze test, we select the ground-truth word from the given options instead of the whole vocabulary. Secondly, instead of predicting word sentiment polarity, we treat it as prior knowledge to assist in predicting overall sentiment. Thirdly, we fine-tune the tasks with only task-specific data instead of post-training them with large-scale domain-specific corpora. Fourthly, we do not substitute any element of the input with "MASK" identifiers.

3 Methodology

Figure 1 illustrates the framework of KESA. In order to promote the main task, two straightforward auxiliary tasks are proposed. It is motivated by Hebbian theory, which claims that the cells that fire together wire together (Hebb, 2005). For instance, when painting and eating together, the neurons activated by painting and food will be easier to connect. After some time, the nerves stimulated by food and painting will be activated simultaneously, making the latter more pleasant. The first task is like the word cloze test, where the correct sentiment word is necessary to be selected among the options. The second task is a more approachable version of the main task, where sentiment at the word-level is provided to help infer sentiment at the sentence-level. We believe that facilitating the challenging task with easier tasks, and then the challenging task may be easier. In addition, to unify several types of labels into a single label, we investigate two kinds of label combination methods. In the subsequent subsections, we will detail the two proposed auxiliary tasks and label combination methods. For convenience, we first give some notations used in the following subsections.

Formally, let $L = \{l_1, l_2, \cdots, l_M\}$ denote the sentiment lexicon with M sentiment words, and $S = \{w_1, w_2, \cdots, w_N\}$ denote an input sentence of length N. $P_S \in C$, $P_w \in Z$ represent the sentiment polarity of sentence S and sentiment word
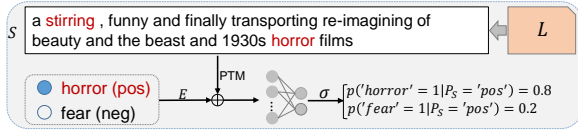
3

Figure 2: A demonstration of auxiliary task A. The sentence is sampled from SST2 dataset, E refers to word embedding table, and σ refers to the Softmax layer. It shows that when the polarity of the sentence is "positive", the probability of "horror" falling within the sentence is 0.8.
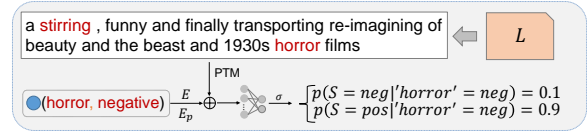

Figure 3: A demonstration of auxiliary task B. The sentence is sampled from SST2 dataset, E and E_p refer to word/polarity embedding table, respectively, and σ refers to the Softmax layer. It means that when the polarity of "horror" is "negative", the probability of sentence S being "negative" is 0.1.

w, respectively. C means all the sentence sentiment labels, and Z represents the word sentiment set. $Y_{w,S} \in \{0, 1\}$ represents the ascription relationship between word w and sentence S, where and $Y_{w,S} = 1$ means that w belongs to 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 PTMs to get the context state $h_{[\text{CLS}]}$. Then the context state is fed into a linear layer and a Softmax layer to get the probability $\hat{P}_S$ of each sentiment label, i.e., $\hat{P}_S = \text{Softmax}(W_1 h_{[\text{CLS}]} + b_1)$, where W_1 and b_1 are the model parameters.

3.2 Task A: Sentiment Word Cloze

Existing sentiment word prediction tasks replace identified sentiment words with "MASK" identifiers in the input, and then reconstruct them in the output layer. In this process, the probability distribution over the vocabulary of PTMs is computed. It is computationally expensive, take RoBERTa-base as an example, the size of its vocabulary is $50,265$. Besides, replacing sentiment words with "MASK" may change the overall sentiment semantics of the input. To alleviate the above issues, sentiment word cloze is designed, which aims to reduce the computational cost, i.e., the number of parameters.

Given a training sample (S, P_S), we first recognize all the sentiment words in S according to the sentiment lexicon by exact word match. Then, we choose one of them as sentiment word w_i and record its sentiment polarity as P_{w_i}. Meanwhile, we randomly sample one sentiment word from the sentiment lexicon as w_j and record its sentiment polarity as P_{w_j}. Next, S is fed into PTMs and its first token ([CLS]) representation $h_{[\text{CLS}]}$ is used as sentence representation. Meanwhile, we extract the

embeddings of the sentiment word w_i and w_j as e, and the embeddings of its sentiment polarity p_{w_i} and p_{w_j} as e'. Then a linear layer and a Softmax layer is used to compute each label's probability,

$$\hat{O}_1 = \text{Softmax}(W_2(h_{[\text{CLS}]} + e + e') + b_2) \quad (1)$$

where W_2 and b_2 are model parameters and we will detail them in the subsequent subsection.

SWC learns the influence of overall sentiment of the sentence (global information) on sentiment words within it (local information). Figure 2 gives an example of the procedure of SWC. In this example, "stirring", "funny", "beauty" and "horror" are first recognized as sentiment words. "horror" is then randomly selected as the correct option, and "fear" is randomly sampled as a false option. The sentence S is input into PTMs to get the context state $h_{[\text{CLS}]}$. Meanwhile, the word embeddings of "horror" and "fear" are lookup from the word embedding table E. Correspondingly, their sentiment embeddings are looked up from sentiment embedding table E_p, respectively. Subsequently, $h_{[\text{CLS}]}$ is concatenated with the word and sentiment embeddings of the two options, respectively, to produce sentiment enhanced or polluted sentence representation. Finally, the SWC task is required to distinguish between the enhanced and polluted sentence representation.

3.3 Task B: Conditional Sentiment Prediction

Existing word polarity prediction tasks replace sentiment words with "MASK" in the input, and recover their sentiment labels in the output layer. In this process, sentiment words and their sentiment labels are extracted by sentiment lexicon or statistical methods, and they may be inaccurate. To alleviate the above issues, conditional sentiment prediction is designed, which aims to reduce the dependence on the accuracy of sentiment lexicon.

More specifically, given a training sample

(S, P_S), similar to SWC, we first choose one sentiment word w from all sentiment words in S recognized with the sentiment lexicon, meanwhile recording its sentiment polarity P_w. After that, sentiment word embedding e_w and its polarity embedding e'_w are lookup from the embedding table and polarity embedding table, respectively. Next the input sentence S is fed into PTMs to get the context state h_{CLS}. Afterwards, we concatenate e_w, e'_w and $h_{\text{[CLS]}}$ to enhance sentence representation with sentiment word and its sentiment polarity, then pass them through a linear layer and a Softmax layer to predict the probability, i.e.,

$$\hat{O}_2 = \text{Softmax}(W_3(h_{\text{[CLS]}} + e_w + e'_w) + b_3) \quad (2)$$

where W_3, b_3 are model parameters and we will detail them in the next subsection. CSP learns the influence of the sentiment polarity of a word 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

Both auxiliary tasks contain multiple kinds of labels. Specifically, for the SWC 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, both overall sentiment P_S and sentiment polarity P_w of a word are involved. Intuitively, multiple kinds of labels can describe the input sentence from different perspectives. Therefore, encouraging the model to leverage different helpful information simultaneously and improving generalization performance (Caruana, 1997).

To treat the various kinds of labels in a uniform manner, we propose 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 conditional combination motivated by Lee et al. (2020), which models the conditional probability distribution of multiple kinds of labels. This method essentially predicts one kind of label with other kinds of labels as prior conditions.

Joint combination. For task A (SWC), given the overall logits $\hat{O}_1$ in Eq. 1, we need to predict the joint probability distribution of the word ascription label Y and the sentence polarity P_S. That is, $p(Y, P_S | \hat{O}_1) \in \mathbb{R}^{|Y| \times |C|}$, where $|Y|$ means

the number of Y's labels ($\{0, 1\}$) and $|C|$ means the number of P_S's labels, e.g., ({`positive`, `negative`}). For task B (CSP), given the overall logits $\hat{O}_2$ in Eq. 2. Similarly, we need to predict the joint distribution of the word polarity P_w and the sentence polarity P_S. That is, $p(P_w, P_S | \hat{O}_2) \in \mathbb{R}^{|Z| \times |C|}$, where $|Z|$ means the number of P_w's labels ({`positive`, `negative`}).

Conditional combination. For task A, given the overall logits $\hat{O}_1$ in Eq. 1, we predict the probability to each word ascription label Y under the condition that sentence polarity P_S is known, i.e., $p(Y | \hat{O}_1, P_S) \in \mathbb{R}^{|Y|}$. To get this, we simply choose the according logits indexed by P_S from $\hat{O}_1$ followed by normalization. Similarly, For task B, given the overall logits $\hat{O}_2$ in Eq. 2, the conditional probability of sentence sentiment polarity P_S given the word sentiment polarity P_w is $p(P_S | \hat{O}_2, P_w) \in \mathbb{R}^{|C|}$. For that, we just select the according logits indexed by P_w from $\hat{O}_2$ followed by normalization.

3.5 Loss Function

We take cross entropy as our loss function, which is a standard selection in classification problem. The loss function is defined as the cross-entropy between the predicted probability $\hat{P}_S$ 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 $\hat{P}_S$ is weighted by $\hat{O}_1, \hat{O}_2$:

$$W_4(p(P_S | \hat{O}_1, Y) \,||\, p(P_S | \hat{O}_2, P_w)) \in \mathbb{R}^C \quad (4)$$

where $W_4 \in \mathbb{R}^{2 \times 1}$ is model parameters, $||$ refers to concatenation, $p(P_S | \hat{O}_1, Y)$ and $p(P_S | \hat{O}_2, P_w)$ are extracted from $\hat{O}_1$ and $\hat{O}_2$ indexed by Y and P_w, respectively. Note that, we omit the bias 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. $\gamma > 0.0$ to ensure that the parameters of the auxiliary tasks can be optimized by back propagation. $\gamma < 1.0$ to prevent the final loss is dominated by the auxiliary task loss and diminishing the performance of the main task (Liu et al., 2019a).

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.

4 Experiment

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 effectiveness of the proposed method for sentence-level sentiment analysis, we compare our method with two types of competitive baselines, including popular vanilla pre-trained models (PTMs) 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 PTMs.

Sentiment Knowledge Enhanced Post-trained Language Models. We also adopt some methods focusing on leveraging sentiment knowledge, two of the influential methods are used as baselines, i.e., SentiLARE (Ke et al., 2020) and SentiX (Zhou et al., 2020). Both design the sentiment word prediction task and the word polarity prediction task. More precisely, the sentiment word is first idenfied and masked, then the PTMs are prompted to recover the corresponding masked words and their corresponding sentiment information. Second, both continue pre-training vanilla PTMs on million scale domain-specific corpora, i.e., Yelp Dataset Challenge 2019 for SentiLARE , Yelp Dataset Challenge 2019 and Amazon review dataset for SentiX. In terms of PTMs, 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 PTMs on sentiment analysis, of which two auxiliary tasks are designed, i.e., SWC and CSP. However, the difference between KESA and SKEP, SentiLARE, SentiX arises from the following. First, the number of options is much smaller than the size of vocabulary of the PTMs. Second, word sentiment is used as local prior information rather than the ground-truth label. Third, no extra corpora are used, and auxiliary tasks are integrated into fine-tuning instead of post-training phase. Fourth, sentiment words are not replaced with "MASK" identifiers.

4.3 Sentiment Lexicon

We extract word sentiments from SentiWordNet 3.0 (Baccianella et al., 2010). Notably, each word in SentiWordNet 3.0 has several usage frequency levels and is linked with different semantic and sentiment scores. Intuitively, we set the sentiment polarity of a word according to its most vital 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). Therefore, we set the polarity of "thirsty" to "negative", considering it has a larger weight of "negative".

4.4 Implementation Details

We implement our model using *HuggingFace's Transformers*[1]. 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 PTM. 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. To explore the influence of auxiliary task on the main task, we search the loss balance weight γ from $\{0.01, 0.1, 0.5, 1.0\}$. These weights are tested based on the following considerations. First, the weights in $(0.0, 1.0)$ should be tested evenly. Second, we argue that higher auxiliary task weights may dominate the total loss. On the contrary, smaller weights should be better, and 0.01 is

---

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

6

selected. We fine-tune each model for 3 epochs, and the best checkpoints on the development set are used for inference. As for each dataset, with a reproducible implementation, we run 4 times with different random seeds, and the average results are reported. Moreover, to make a fair comparison, all methods use the same seeds for the same dataset.

4.5 Overall Results

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‡	94.96‡	59.26	95.83**

Table 3: Overall accuracy on sentence-level sentiment classification benchmarks. The marker # means that the results are reported in the original paper while − means no reported results. The marker * refers to our re-implementation. The markers ** and ‡ indicate that our model significantly outperforms the best baselines with t-test, p-value < 0.01 and 0.05, respectively.

Table 3 reports the results of our method and all baselines, 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. We find that KESA works across all four datasets, with overall improvements of (0.76%, 0.38%, 0.72%, 0.1%) on (MR, SST2, SST5, IMDB), respectively. Although SentiX and SentiLARE are post-trained on million scale domain-specific corpora. There are still gains when fine-tuning with KESA, indicating that KESA is additive to pre-trained models and sentiment knowledge enhanced post-trained models.

4.6 Ablation Results

The ablation studies of the SWC and CSP task are reported in Table 4. We find our SWC outperforms the baselines by up to 0.7%. The results verify the correctness of our motivation and the effectiveness of the word ascription label being supervised signal. This is probably because the word ascription label pushes the model to focus on the interactions between sentence sentiment and its items, and this kind of connection between global information and local information can promote the main task. Likewise, we also report the results of the CSP task

Model	MR	SST2	SST5	IMDB
BERT*	**86.62**	91.38	53.52	93.45
+SWC	86.30	91.46	54.21	**93.59**
+CSP	86.45	**91.70**	**54.38**	93.51
+KESA	86.29	91.56	54.13	93.51
XLNet*	88.83	92.75	54.95	94.99
+SWC	89.05	**93.47**	55.51	**95.03**
+CSP	**89.31**	92.79	55.45	94.97
+KESA	89.10	93.01	**55.94**	95.00
RoBERTa*	89.84	94.00	57.09	95.13
+SWC	89.81	94.22	57.22	95.40
+CSP	89.86	94.17	**57.24**	95.44
+KESA	**90.07**	**94.40**	57.18	**95.46**
SentiX*	86.81	92.23	55.59	94.62
+SWC	87.31	92.20	55.74	**94.71**
+CSP	87.35	92.24	**55.83**	94.61
+KESA	**87.36**	**92.52**	55.78	94.57
SentiLARE*	90.50	94.58	58.54	95.73
+SWC	90.74	94.72	**59.29**	95.80
+CSP	91.10	94.91	58.59	95.80
+KESA	**91.26**	**94.96**	59.26	**95.83**

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

solely. With the addition of CSP, performance is increased on nearly all datasets with a maximum gain of 0.86%. The results demonstrate that adding the sentiment of word explicitly brings more information and lowers the difficulty of the CSP task than that of the main task. Afterward, this similar but easier auxiliary task promotes the optimization for the main task, namely, fire together wire together. Remarkably, the experimental results show that the combination of two auxiliary tasks is not systematically superior to the performance of SWC or CSP used alone. This is likely because SWC learns the influence of sentences on words, while CSP learns the influence of words on sentences, and they may compete with each other in some cases. As reported in (Bingel and Søgaard, 2017), multiple tasks may promote each other or compete with each other (negative learning). Above all, these results remind us that the combinations of multiple tasks need to be carefully analyzed, even if each is effective. Even so, KESA still outperforms the baselines on all evaluated datasets.
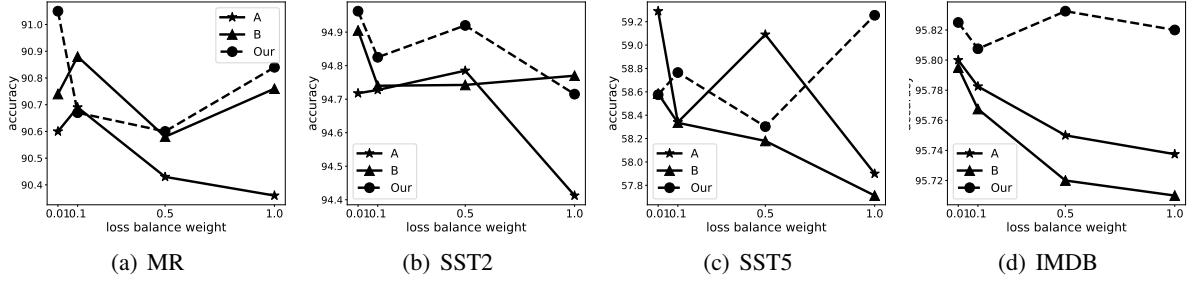
| (a) MR | (b) SST2 | (c) SST5 | (d) IMDB |

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 task A and B are tested solely. Our refers to KESA.

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

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

4.7 Analysis on Loss Balance Weight

We further analyze the impact of loss balance weight, as shown in Figure 4. It can be observed that, generally, lower loss balance weight achieves better performance in most cases. More specifically, take IMDB as an example, as there are more training samples and longer sequence length (512), making it less sensitive to seeds. With the decrease of loss balance weight, the advantages gradually increase on SWC, CSP, and KESA, loss balance weight equal to 0.01 always performs better than 1.0. This is presumably due to that the weight of auxiliary tasks should be a small value to avoid undue impact on the main task.

4.8 Analysis on Label Combination

In terms of unifying several types of labels in each task, we carry out experiments to compare their performance. SentiX and SentiLARE are selected, as they perform better. The result is shown in Table 5. Overall, for both SWC and CSP tasks, the conditional combination is slightly better than the joint combination in most cases across all evaluated datasets. Specifically, the difference is greater upon SentiLARE than that of SentiX. The joint combination is better on MR, SST2, and IMDB except SST5. All the results above demonstrate that the label combination method should be selected based on PTMs and datasets. Nevertheless, we recommend conditional combination as the default.

4.9 Analysis on Parameters

For SWC, the number of increased parameters is $W_2 \in \mathbb{R}^{|Y|d \times |C||Y|}$, $b_2 \in \mathbb{R}^{|C||Y|}$ and polarity embedding $E_p \in \mathbb{R}^{|Z| \times d}$. For CSP, the number of increased parameters is $W_3 \in \mathbb{R}^{d \times |Z||C|}$, $b_3 \in \mathbb{R}^{|Z||C|}$ and polarity embedding $E_p \in \mathbb{R}^{|Z| \times d}$. Besides, 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 number of parameters increase induced by KESA is $W_2, W_3, W_4, b_2, b_3, b_4$ and E_p. In the experiments, $|C| \leq 5$, $|Y| = 2$, $|Z| = 2$, $d = 768$, and $V = 30,522$ (refers to the size of the vocabulary of base BERT). The parameters increased by SWC is about 0.7% (Y/V) of that of recovering from the vocabulary.

5 Conclusion

In this paper, we propose two sentiment-aware auxiliary tasks to include sentiment knowledge in pre/post-trained language models. Further, we propose joint and conditional combinations to unify multiple kinds of labels into a single label. In addition, both auxiliary tasks are integrated into the fine-tuning phase to avoid a large volume of domain-specific data. Finally, sentiment words are not replaced with "MASK" to avoid sentiment information loss. Though straightforward and conceptually simple, KESA still further improves on solid baselines. Our work verifies that more knowledge integrated at the input or output end can help improve the performance of the model.

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

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

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.

Donald Olding Hebb. 2005. *The organization of behavior: A neuropsychological theory*. Psychology Press.

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, and Zhi Yu. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis.

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*

9

Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 115–124.

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.

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.

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.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335.

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

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.

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.

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.