Perturbation-based Self-supervised Attention for Text Classification

Anonymous ACL submission

Abstract

For text classification, the traditional attention mechanisms usually focus too much on frequent words, and need extensive labeled data in order to learn. This paper proposes a perturbation-based self-supervised attention approach to guide attention learning without any annotation overhead. Specifically, we add as much noise as possible to all the words in the sentence without changing their semantics and predictions. We hypothesize that words that tolerate more noise are less significant, and we can use this information to refine the attention distribution. Experimental results on three text classification tasks show that our approach can significantly improve the performance of current attention-based models, and is more effective than existing self-supervised methods. We also provide a visualization analysis to verify the effectiveness of our approach.

1 Introduction

Attention mechanisms (Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017) play an essential role in Natural Language Processing (NLP) and have been shown to be effective in various text classification tasks, such as sentiment analysis (Lin et al., 2017; Tang et al., 2019; Choi et al., 2020), document classification (Yang et al., 2016) and natural language inference (Chen et al., 2017). They achieve significant performance gains, and can be used to provide insights into the inner workings of the model. Generally, the attention learning procedure is conditioned on access to large amounts of training data without additional supervision information.

Although the current attention mechanisms have achieved remarkable performance, several problems remain unsolved. First, learning a good attention distribution without spurious correlations for neural networks requires large volumes of informative labeled data (Barrett et al., 2018; Bao et al., 2018). As described in the work of Wallace et al. (Wallace et al., 2021), after inserting 50 poison examples with the name "*James Bond*" into its training set, a sentiment model will frequently predict a positive whenever the input contains this name, even though there is no correlation between the name and the prediction. Second, attention mechanisms are prone to focus on high-frequency words with sentiment polarities and assign relatively high weights to them (Xu et al., 2018; Li et al., 2018; Tang et al., 2019), while the higher frequency does not imply greater importance.

Especially when there's an adversative relation in a text, some high-frequency words with strong sentiment valence need to be selectively ignored based on the context of the whole text. In these cases, these words will mislead the model because the important words don't get enough attention. The sentences in Figure 1 illustrate this problem. In most training sentences, as shown in the first four rows, "*better*" and "*free*" appear with positive sentiment, which makes the attention mechanism accustomed to attaching great importance to them and relating them to positive predictions. However, the two words are used ironically in the fifth sentence, and the model pays the most attention to them while the critical word – "*leave*" – is not attended to, resulting in an incorrect prediction. Based on these observations, there's reason to believe that the attention mechanisms could be improved for text classification.

To tackle this problem the most direct solution is to add human supervision collected by manual annotation (Zhang et al., 2016; Bao et al., 2018; Camburu et al., 2018) or special instruments (Barrett et al., 2018; Sood et al., 2020b,a; Malmaud et al., 2020) (e.g., eye-tracking), to provide an inductive bias for attention. These approaches are costly, the labeling is entirely subjective, and there is often high variance between annotators. In particular, Sen et al. (Sen et al., 2020) point out that there is a huge difference between machine and

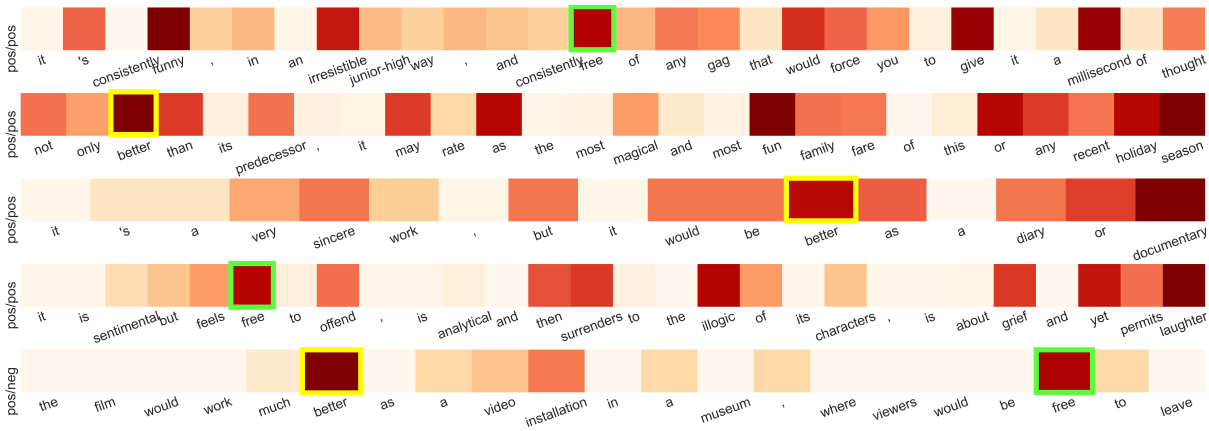

Figure 1: The attention visualization for five sentences. The "A/B" style tags before each row mean the model's prediction is A and the label is B. The first four sentences are selected from training sets as representatives containing high-frequency words - "better" (yellow box) and "free" (green box). The last sentence including both of the two words is selected from testing sets, typically showing that the distribution of attention weights when some words in the sentence appear frequently in the corpus but are unimportant to the current prediction.

human attention and it is difficult to map human attention to machine attention.

Another flexible solution is to measure attribution scores, i.e., how much each token in a text contributes to the final prediction, to approximate an importance distribution as an attention supervision signal (Li et al., 2016; Choi et al., 2019; Tang et al., 2019; Choi et al., 2020). Generally, the attribution scores are obtained by masking each token one by one to generate counterfactual examples, reflecting the difference in the softmax probability of the model after masking each token. These approaches have little or no additional annotation overhead and augment supervision information from the training corpus to refine the attention distribution. Despite their success, masking schemes can give rise to an out-of-distribution (OOD) problem (Hendrycks and Gimpel, 2016; Chang et al., 2018; Yi et al., 2020). That is, the generated counterfactuals deviate from the training data distribution of the target model, resulting in an overestimation of the contribution of unimportant tokens. The OOD problem induced by existing masking schemes makes it difficult to identify whether high-scoring tokens contribute significantly to the prediction. Furthermore, most of them are limited to generating uniform attention weights for the selected important words. Obviously, the contribution of different important words to the model should also be different according to the context, e.g., the word *leave* should have a higher attention weight than *better* and *free* for the fifth sentence in Figure 1.

Some efforts reveal that the output of neural networks can be theoretically guaranteed to be invariant for a certain magnitude of input perturbations through establishing the concept of maximum safety radius (Wu et al., 2020; La Malfa et al., 2020) or minimum disturbance rejection (Weng et al., 2018). In simple terms, these approaches evaluate the minimum distance of the nearest perturbed text in the embedding space that is classified differently from the original text. Inspired by this work, we propose a novel perturbation-based self-supervised attention learning method without any additional annotation overhead for text classification. Specifically, we design an attention supervision mining mechanism called Word-based Concurrent Perturbation (WBCP), which effectively calculates an explainable word-level importance distribution for the input text. Concretely, WBCP tries to concurrently add as much noise as possible to perturb each word embedding of the input, while ensuring that the semantics of input and the classification outcome is not changed. Under this condition, the words that tolerate more noise are less important and the ones sensitive to noise deserve more attention. We can use the permissible perturbation amplitude as a measure of the importance of a word, where small amplitude indicates that minor perturbations of that word can have a significant influence on the semantic understanding of input text and easily lead to prediction error.

According to the inverse distribution of perturbation amplitude, we can get sample-specific attention supervision information. Later, we use this supervision information to refine the attention dis-

tribution of the target model and iteratively update it. Notably, our method is model-agnostic and can be applied to any attention-based neural network. It generates attention supervision signals in a self-supervised manner to improve text classification performance without any manual labeling and incorporates Perturbation-based Self-supervised Attention (PBSA) to avoid the OOD problem caused by the masking scheme. In addition, it can also generate special attention supervision weights adaptively for each sample based on the perturbation amplitude, rather than allocate them uniformly.

In summary, the contributions of this paper are as follows:

(1) Through analysis of current methods, we point out the disadvantages and drawbacks of current attention mechanisms for text classification.

(2) We propose a simple yet effective approach to automatically mine the attribution scores for the input text, and use it as supervision information to guide the learning of attention weights of target models.

(3) We apply our approach to various text classification tasks, including sentence classification, document categorization, and aspect-level sentiment analysis. Extensive experiments and visualization analysis show the effectiveness of the proposed method in improving both model prediction accuracy and robustness.

2 Related work

Work related to our method can be categorized into three types: Introducing human attention; using external resources or tools; and using self-supervision.

Introducing human attention Adding human supervision to attention has been shown to effectively alleviate attention bias and improve model prediction accuracy on a range of tasks (Zhang et al., 2016; Camburu et al., 2018; Sood et al., 2020b,a; Malmaud et al., 2020). In general, the annotators need to explicitly highlight the important words or rationales (Zhang et al., 2016; Bao et al., 2018; Camburu et al., 2018) for the given sample. Obviously, the annotation is very labor-intensive and expensive in real-world scenarios, so an alternative is to use implicit signals such as eye gaze (Barrett et al., 2018; Sood et al., 2020b,a; Malmaud et al., 2020). For these methods, it is expected that the model can generate similar attention to human supervision. However, human recognition and model reasoning processes may be inconsistent (Jacovi and Goldberg, 2020), and aligning the two is challenging (Sen et al., 2020).

Using external resources or tools With the development of NLP, many corpora and tools, such as Dependency Tree and Synonym Dictionary, are created to obtain a deeper understanding of words and sentences. Therefore, some methods (Kamigaito et al., 2017; Zou et al., 2018; Nguyen and Nguyen, 2018; Zhao et al., 2020) that generate attention supervision information according to existing corpora and tools emerge. For example, Nguyen et al. (Nguyen and Nguyen, 2018) introduce attention supervision information based on important words selected by semantic word lists and dependency trees. Similarly, Zhao et al. (Zhao et al., 2020) first train the model on the document-level sentiment classification and then transfer the attention knowledge to a fine-grained one for aspect-level sentiment classification. However, these methods still rely on annotations based on parsers or external resources, and the performance depends heavily on the quality of the parser.

Self-supervised attention learning Currently, self-supervised attention learning frameworks (Li et al., 2016; Choi et al., 2019; Tang et al., 2019; Choi et al., 2020; Su et al., 2021) have become the mainstream method because they do not require additional annotation overhead. They usually mask or erase each token one by one and quantify the difference in predictions of the model after masking each token, to approximate an importance distribution as attention supervision information. For example, Tang et al. (Tang et al., 2019) divide the words in sentences into the active set and the misleading set by progressively masking each word with respect to the maximum attention weight, and augment them to make the model focus on the active context words. Similarly, Choi et al. (Choi et al., 2020) adopt the masking method to find the unimportant words and gradually reduce their weights. These methods use a self-supervised paradigm to mine important words, which can greatly reduce the annotation cost and improve the robustness of the model. Nevertheless, the masking scheme they follow has an OOD problem. The counterfactuals generated by the mask operation deviate from the original training set distribution, which easily leads to the over-evaluation of unimportant words. In addition, the above methods usually assign the

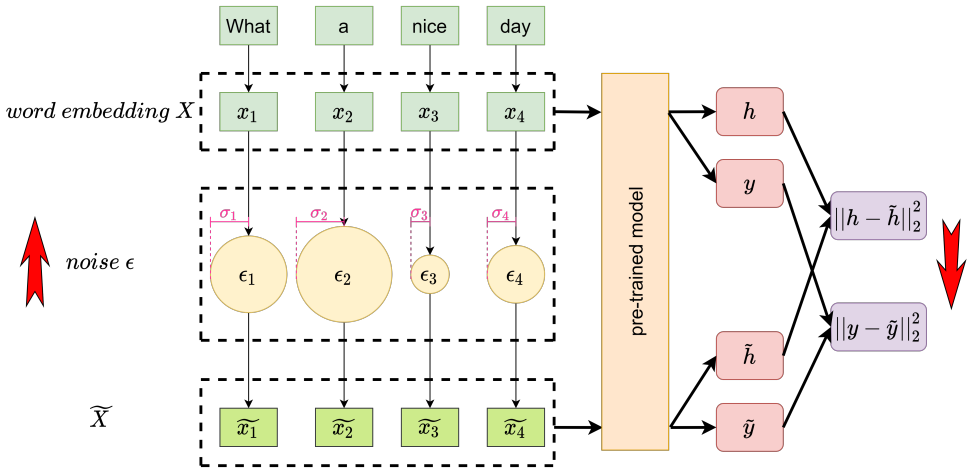

Figure 2: The diagram of WBCP. The left part of the figure corresponds to the last term of Eq. (3), which illustrates the process of adding noise that follows a Gaussian distribution to each word. The right part of the figure corresponds to the first two terms of Eq. (3), indicating the constraint of trying to not change the semantics and predictions after the noise is introduced.

same weight to the extracted important words, but in our opinion, different words should have different contributions to the classification.

3 Proposed method

In this section, we propose a Perturbation-based Self-supervised Attention (PBSA) mechanism to enhance the attention learning process and provide a good inductive bias. We first design a Word-based Concurrent Perturbation (WBCP) to automatically mine the attribution score for each word and use this as a measure of its degree of importance. Then we use the measure mentioned above to compute a word-level importance distribution as supervision information. Finally, we describe how to use the supervision information to refine the attention mechanism of the target model, improving the accuracy and robustness of text classification tasks.

3.1 Word-based Concurrent Perturbation

The basic assumption of our design is based on the following fact: under the premise of trying not to change the semantics of the input text, unimportant words can withstand more changes than more significant ones. Specifically, a little noise on keywords can lead to dramatic changes in the final results, while greater noise on the unimportant ones won't easily lead to changes in results. Therefore, we can estimate the importance distribution of the words according to the maximum amount of noise they can tolerate. To be specific, we try to concurrently add as much noise as possible to perturb each word embedding without changing the latent representations (e.g., the hidden states for classification) of the text and the prediction result. The above process can be optimized according to the maximum entropy principle.

Given a sentence consisting of n words $s = \{w_1, w_2, ..., w_n\}$, we map each word into its embedding vector $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$. Here we assume that the noise on word embeddings obeys a Gaussian distribution $\boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_i = \sigma_i^2 \mathbf{I}\right)$ and let $\widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$ denote an input with noise $\boldsymbol{\epsilon}_i$. We use $\boldsymbol{h}$, $\boldsymbol{y}$ and $\widetilde{\boldsymbol{h}}$, $\widetilde{\boldsymbol{y}}$ to indicate the hidden state for classification and the prediction result of a pre-trained model with no noise and with noise respectively. Then we can write the loss function of WBCP as follows:

$$\begin{aligned} \mathcal{L}_{WBCP} = &||\widetilde{\boldsymbol{h}} - \boldsymbol{h}||_2^2 + ||\widetilde{\boldsymbol{y}} - \boldsymbol{y}||_2^2 \\ &-\lambda \sum_{i=1}^{n} H(\boldsymbol{\epsilon}_i)|_{\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i = \sigma_i^2 \mathbf{I})}, \end{aligned} \tag{1}$$

where λ is a hyperparameter that balances the strength of noise. The first and the second term of Eq. (1) mean that we need to minimize the L2-normalized euclidean distance between the two hidden states and between the two predictions respectively, to quantify the change of information (Jain and Wallace, 2019). The first term maintains latent representations to prevent modification of the text semantics, and the second term prevents excessive perturbations from causing the model to mispredict. The last term indicates that we need to maximize the entropy $H(\boldsymbol{\epsilon}_i)|_{\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i = \sigma_i^2 \mathbf{I})}$ to encourage adding as much noise as possible to each word embedding. We can simplify the maximum entropy

of the Gaussian distribution as follows (the detailed formula derivation is listed in Appendix A):

$$Maximize(H(\boldsymbol{\epsilon}_i)) \Rightarrow Maximize(\log \sigma_i) \quad (2)$$

Finally we can use Eq. (2) to rewrite our final objective function:

$$\begin{aligned}\mathcal{L}_{WBCP} = ||\widetilde{\boldsymbol{h}} - \boldsymbol{h}||_2^2 + ||\widetilde{\boldsymbol{y}} - \boldsymbol{y}||_2^2 \\ +\lambda \sum_{i=1}^{n} \log(-\sigma_i)\end{aligned} \quad (3)$$

The illustration of WBCP is given in Figure 2. After fixing the parameters of the pre-trained model, the only learnable parameters $\sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ can be considered as the perturbation radii, which is positively associated with the perturbation amplitude. Specifically, the larger σ_i WBCP gets, the more likely $\boldsymbol{\epsilon}_i$ is a big number, the more noise is added to $\boldsymbol{x}_i$, and the less important it is. As what is shown in the picture, it is obvious that $\sigma_2 > \sigma_1 > \sigma_4 > \sigma_3$. According to the analysis listed above, we know that $\boldsymbol{w}_2$ (*a*) is the least important word and $\boldsymbol{w}_3$ (*nice*) is the most significant one, for $\boldsymbol{x}_2$ can tolerate the most noise while $\boldsymbol{x}_3$ can hardly stand any perturbation.

3.2 Attention supervision

We obtain the σs, the perturbation magnitudes, by optimizing Eq. (3) on the pre-trained model. If a word embedding $\boldsymbol{x}_i$ can tolerate more noise without impacting the semantics of input text, σ_i will be larger, which means the word $\boldsymbol{x}_i$ is less important. Conversely, small σ_i indicates that slight perturbations of word embedding $\boldsymbol{x}_i$ will lead to semantic drift and may affect the classification result. We can therefore use the perturbation magnitude to compute a word-level importance distribution as attention supervision information, as shown below:

$$\begin{aligned}\alpha_i' &= 1 - \frac{\sigma_i}{\max_j\{\sigma_j\}} \\ \widetilde{\boldsymbol{\alpha}} &= \text{Softmax}(\boldsymbol{\alpha'})\end{aligned} \quad (4)$$

It is worth noting that our method generates sample-specific attention supervision, where the weight of each word is quantified according to the perturbation magnitude, instead of using the same importance weight for all words (Tang et al., 2019; Choi et al., 2020). Also, the quantification occurs in the embedding space rather than replacing the token with a predefined value, thus avoiding the OOD problem caused by masking schemes.

Algorithm 1: Perturbation-based self-supervised attention

Input: training dataset D, attention-based model $f(\cdot, \theta)$, the number of iterations T.

Pre-train model $f(\cdot, \theta)$ on D and update θ using Adam.

for $t = 1, ...T$ **do**

- Fix θ, and minimize WBCP objective function by Eq. (3) using Adam.
- Obtain the perturbation amplitude σ for each sample in D.
- Calculate the attention supervision $\widetilde{\alpha}$ by Eq. (4) for each sample in D.
- Re-train model on D with the attention supervision $\widetilde{\alpha}$ by Eq. (5) and update θ using Adam.

end

3.3 Perturbation-based Self-supervised Attention

We do not use $\widetilde{\boldsymbol{\alpha}}$ to generate a new attention distribution to replace the original one $\boldsymbol{\alpha}$. Rather, we use it as a supervision target for the attention weights. We want the attention supervision to make the model notice more words that have an influence on the output. In this way, some low-frequency context words with great importance that would normally be ignored can be discovered by attention learning. In this section, we describe how to exploit the supervision information $\widetilde{\alpha}$ to guide the learning of model attention strengths.

Our method is shown in Algorithm 1. We first pre-train an attention-based model $f(\cdot, \theta)$ based on the classification dataset D. We then fix the model parameters θ and minimize the WBCP objective using Eq. (3) to obtain the perturbation amplitude σ for each sample, and used to compute the attention supervision $\widetilde{\alpha}$ using Eq. (4). We then retrain the model using $\widetilde{\alpha}$ to guide the attention distribution α produced by the model. The above process can iterate T times to capture the important distribution more accurately. The training objective function with attention supervision $\widetilde{\alpha}$ is defined as follows:

$$\begin{aligned}\mathcal{L}_{cls} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m \log y_m \\ +\gamma \text{KL}(\widetilde{\alpha}_m || \alpha_m),\end{aligned} \quad (5)$$

where M is the number of samples, γ is a hyper-parameter that controls the strength of attention

Table 1: Experimental accuracy on the document-level and sentence-level classification for Att-BiLSTM.

Model	IMDB	SST2	TREC	MR	CR	SUBJ	MPQA	Average
Att-BiLSTM	87.21	83.42	90.60	77.04	76.82	89.82	70.59	82.20
Att-BiLSTM+Gradient	86.79	85.06	91.20	77.60	76.54	89.82	70.76	82.53
Att-BiLSTM+PBSA	**89.14**	**85.72**	**92.20**	**79.05**	**77.64**	**90.53**	**71.31**	**83.65**
Att-BERT	92.53	91.43	96.60	79.26	89.06	94.30	89.69	90.41
Att-BERT+PBSA	**92.61**	**91.93**	**97.20**	**79.97**	**89.38**	**94.76**	**90.21**	**90.86**
BERT	92.92	91.71	96.60	85.47	89.42	96.30	89.59	91.71
BERT+PBSA	**93.48**	**92.20**	**97.80**	**86.08**	**90.21**	**97.50**	**90.57**	**92.54**

supervision, $\hat{y}_m$ and y_m are the ground-truth label and predicted output for the m-th sample respectively. The first term is the Cross-Entropy Loss for classification, and the second term is the Kullback–Leibler Divergence between the distributions of attention α_m produced by model and attention supervision information $\widetilde{\alpha}_m$ for the m^{th} sample.

4 Experiments

We tried PBSA on several text classification tasks, including sentence classification, document categorization, and aspect-level sentiment analysis. Experimental results demonstrate that PBSA consistently enhances the performance and robustness of various attention-based baselines, and outperforms some strong models following self-supervised attention learning. Furthermore, a visualization analysis confirms that our model is capable of generating high-quality attention for target tasks. We aim to answer the following questions: 1. Does PBSA improve model accuracy? 2. Is PBSA more effective than other approaches? 3. How do hyperparameters affect the results? 4. How does PBSA work?

4.1 Datasets and Setup

The statistics of the datasets we use for different tasks are listed in Appendix B. We use a grid search to find the optimal hyperparameters γ and T for each dataset, from the sets $\gamma \in \{0.05, 0.1, 1.0, 2.0, 10, 100\}$ and $T \in \{1, 2, 3, 4\}$. The hyperparameter λ is set to 0.1, the batch size is set to 64. We use the Adam optimizer with learning rate 0.001. The other details of our experiments are listed in Appendix B. We describe all of our baselines in Appendix D.

4.2 Sentence-level and Document-level Classification

To verify that PBSA can improve the performance of the attention-based model, in this section we use the classic Att-BiLSTM (Zhou et al., 2016) and the pre-trained BERT model (Devlin et al., 2018) as the baselines. It is worth noting that BERT uses multiple-head attention, so how to select the suitable head as the supervised target is difficult (Su et al., 2021). Hence, how to effectively combine its multi-head attention with our method is an interesting and valuable question.

We explore two simple strategies to combine our approach with BERT. 1) We first add a scaled dot-product attention layer to the output of BERT to derive a fixed sized sentence representation for classification, and we call this model Att-BERT for short. 2) We also try a simple but effective way to combine the internal multi-head attention in Transformer with our method. Specifically, we average the multi-head attention of all the layers and compress the attention matrix to a vector to be guided by our mechanism. The illustrations of different baselines are shown in Appendix D.1 and Appendix D.2.

Table 1 reports the experimental results on the seven datasets of sentence classification and document categorization. We observe that our method consistently helps improve the accuracy of the baseline on all the datasets. The average accuracy of our approach on the three baselines across seven datasets are 83.65, 90.86 and 92.54, an improvement of 1.44%, 0.45% and 0.83% over the baselines (82.21, 90.41 and 91.71). The results demonstrate that our approach delivers significant performance improvements over the baselines. It also indicates that the current model limits the potential of attention mechanisms when without any supervision information. However, PBSA can mine the potential important words and then guide the attention mechanism of the model to learn a good inductive bias.

Table 2: Experimental results on aspect-level tasks compared with others.

Models	REST		LAPTOP		TWITTER	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
MN (Wang et al., 2018)	77.32	65.88	68.90	63.28	67.78	66.18
MN (Ours)	79.89	65.89	72.68	61.97	68.34	66.23
+Gradient (Serrano and Smith, 2019)	76.85	60.06	71.11	63.53	67.77	64.91
+AWAS (Tang et al., 2019)	78.75	69.15	70.53	65.24	69.64	67.88
+Boosting (Su et al., 2021)	77.66	66.23	69.28	64.17	68.14	67.12
+Adaboost (Su et al., 2021)	76.77	62.29	67.88	60.52	66.96	65.09
+PGAS (Su et al., 2021)	78.98	69.42	70.84	65.58	69.78	67.80
+PBSA	**83.98**	**70.84**	**75.75**	**67.21**	**72.10**	**69.64**
BERTABSA	79.80	71.37	79.38	75.69	76.01	74.52
+PBSA	**79.89**	**71.59**	**79.51**	**75.87**	**76.11**	**74.69**
Att-BERTABSA	83.29	75.87	77.98	75.02	73.99	71.23
+PBSA	**83.41**	**76.70**	**78.65**	**75.53**	**74.45**	**72.88**

4.3 Aspect-level Sentiment Analyis

To further verify the effectiveness of our approach, we apply PBSA into MN (Tang et al., 2016; Wang et al., 2018), BERTABSA (Dai et al., 2021), and Att-BERTABSA (Su et al., 2021). Both BERTABSA and Att-BERTABSA are typical and simple ways to apply BERT to aspect-level classification tasks. The difference is that BERTABSA directly uses the hidden states of the aspect words to classify, while Att-BERTABSA adds an attention layer to the output of BERT. To show that our method truly improves the results, we only use the most critical parts of the model without any other tricks or mechanisms (e.g. the gating mechanism). All the illustrations of baselines are shown in Appendix D.3, Appendix D.4 and Appendix D.5. The results are shown in Table 2. We conduct experiments on three benchmark datasets of aspect-based sentiment analysis and PBSA outperforms all the baselines on all datasets both in accuracy and Macro-F1. Compared with other tasks, PBSA has a more significant improvement on these small-scale datasets, indicating that the original attention lacks a good inductive bias due to limited labeled data. With the help of PBSA, the robustness of the model can be improved effectively.

Table 3: Experimental accuracy on document-level and sentence-level tasks compared with others

Model	SANA (Choi et al., 2020)	PBSA
IMDB	88.03	**89.14**
SST2	84.35	**85.72**

4.4 Comparison with other methods

On the tasks listed above, we compare our method with other advanced self-supervised attention learning approaches. SANA (Choi et al., 2020) generates counterfactuals by a masking scheme and measures the difference in the softmax probability of the model between the counterfactual and original sample as an indicator of important words. AWAS (Tang et al., 2019) and PGAS (Su et al., 2021) progressively mask the word with the largest attention weight or partial gradient. Most of these works don't publish their critical code and do their experiment only on certain specific tasks, so we directly compare our algorithm with their best results published on different tasks respectively. On the document-level and sentence-level tasks (Table 3), PBSA is superior to SANA by 1.11% and 1.37%, which verifies that the word-based concurrent perturbation can mine the importance distribution of words more accurately than the masking scheme. On the aspect-level task (Table 2), compared with AWAS and PGAS, our method improves the model more. As we mentioned in the Introduction (Section 1), our method can generate word-specific attention supervision while others treat the important words equally without discrimination. We speculate that this may be one of the main reasons for our improvement.

From the aspect of human intuition, the gradient-based methods and leave-one-out methods are usually used to improve the interpretability of model. The current self-supervised attention learning methods are mostly based on word masking, which can be seen as a variation of leave-one-out methods. We also try to use the gradient-based method (Serrano and Smith, 2019) to generate supervision information. As shown in Table 1 and Table 2, the gradient-based method does badly on most of the datasets, especially on aspect-level datasets. These results demonstrate that although the gradient-based method can improve the interpretability of the model, it does not necessarily improve the per-

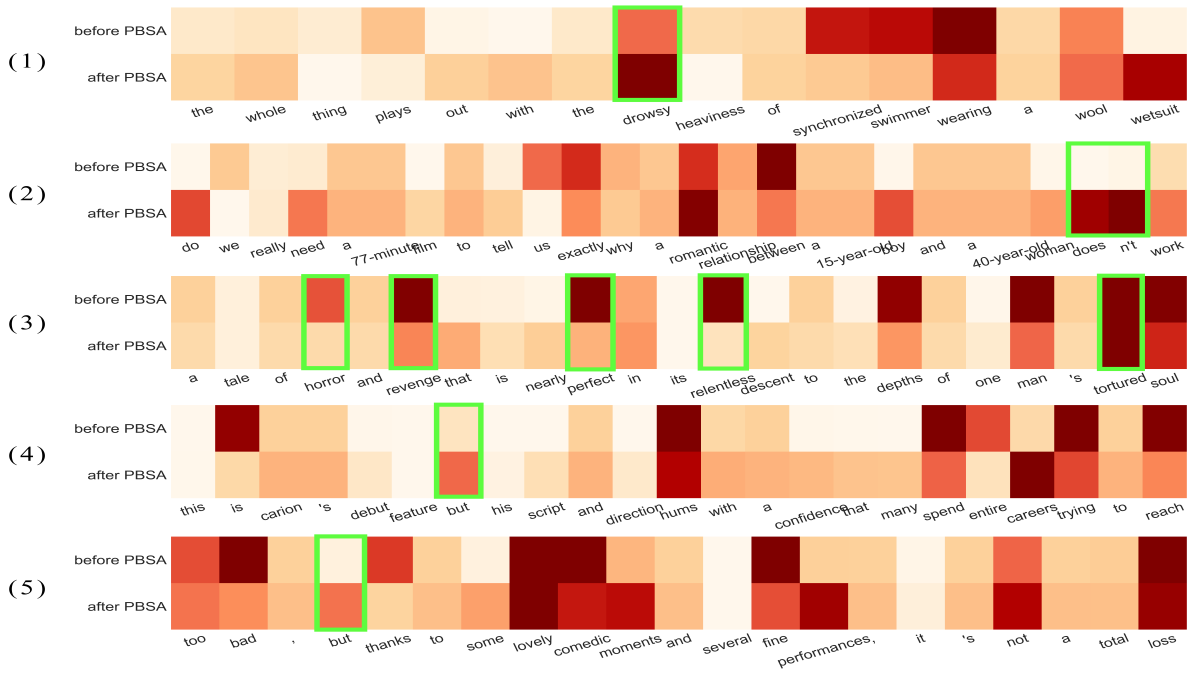

Figure 3: The visualization result of several samples on SST2 test set.

formance. However, our method enhances interpretability while also improving its performance.

4.5 Hyperparameter sensitivity

As shown in Figure 4, our method achieves the best results on REST and TWITTER when $T = 2$ and $T = 1$ respectively. With the increase of T, the performance increases initially, and then decreases due to over-fitting. In practice, we find that one iteration achieves promising results. The hyperparmeter λ controls the perturbation degree of WBCP. When λ is too large, performance is deteriorated due to injecting too much noise. In all of our experiments, we set λ to 0.1. The hyperparmeter γ controls the strength of attention supervision. When γ is too large, it easily leads to overly penalizing the alignment between the model attention and perturbation attention, which may hurt the model's internal reasoning process.

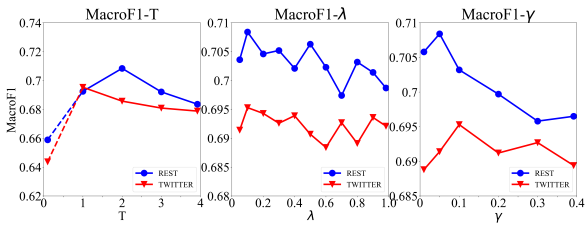

Figure 4: The chart of the fluctuations of Macro-F1 when we change the values of hyperparameters.

4.6 Visualization analysis

As shown in Figure 3, we see that PBSA makes the model pay more attention to important but low-frequency words, reduces the focus on high-frequency words that do not affect the results, increases the difference in weight between words with conflicting meanings, and increases sensitivity to adversative relations in sentences. The detailed explanation is listed in Appendix C.

5 Conclusions and future work

In this paper, we propose a novel self-supervised attention learning method based on word-based concurrent perturbation. The algorithm adds as much as noise to each word in the sentence under the premise of unchanged semantics to mine the supervision information to guide attention learning. Our experiments demonstrate that our method achieves significant performance improvements over the baselines on several text classification tasks. Moreover, we use several visualization samples to interpret how our method guides the internal reasoning process of models. We will try to incorporate our method into more NLP tasks in the future.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly

learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. 2018. Deriving machine attention from human rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1903–1913.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: natural language inference with natural language explanations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9560–9572.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Seungtaek Choi, Haeju Park, and Seung-won Hwang. 2019. Counterfactual attention supervision. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1006–1011. IEEE.

Seungtaek Choi, Haeju Park, Jinyoung Yeo, and Seung-won Hwang. 2020. Less is more: Attention supervision with counterfactuals for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6695–6704.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? A strong baseline for aspect-based sentiment analysis with roberta. *CoRR*, abs/2104.04986.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 49–54.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. 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 3543–3556.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 7–12.

Emanuele La Malfa, Min Wu, Luca Laurenti, Benjie Wang, Anthony Hartshorn, and Marta Kwiatkowska. 2020. Assessing robustness of text classification through maximal safe radius computation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2949–2968.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 946–956.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Jonathan Malmaud, Roger Levy, and Yevgeni Berzak. 2020. Bridging information-seeking human gaze and machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 142–152.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality arxiv : 1310 . 4546v1 [cs . cl] 16 oct 2013.

Minh Nguyen and Thien Huu Nguyen. 2018. Who is killed by police: Introducing supervised attention for hierarchical lstms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2277–2287.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

C. Peng, Z. Sun, L. Bing, and Y. Wei. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *SemEval 2014*, page 27.

Cansu Sen, Thomas Hartvigsen, Biao Yin, Xiangnan Kong, and Elke Rundensteiner. 2020. Human attention maps for text classification: Do humans and neural networks focus on the same words? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4596–4608.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

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.

Ekta Sood, Simon Tannert, Diego Frassinelli, Andreas Bulling, and Ngoc Thang Vu. 2020a. Interpreting attention models with human visual attention in machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 12–25.

Ekta Sood, Simon Tannert, Philipp Mueller, and Andreas Bulling. 2020b. Improving natural language processing tasks with human gaze-guided neural attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 6327–6341.

Jinsong Su, Jialong Tang, Hui Jiang, Ziyao Lu, Yubin Ge, Linfeng Song, Deyi Xiong, Le Sun, and Jiebo Luo. 2021. Enhanced aspect-based sentiment analysis models with progressive self-supervised attention learning. *Artificial Intelligence*, 296:103477.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 557–566.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online. Association for Computational Linguistics.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 957–967.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2020. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807:298–329.

Hengru Xu, Shen Li, Renfen Hu, Si Li, and Sheng Gao. 2018. From random to supervised: A novel dropout mechanism integrated with global information. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 573–582.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jihun Yi, Eunji Kim, Siwon Kim, and Sungroh Yoon. 2020. Information-theoretic visual explanation for black-box classifiers. *arXiv preprint arXiv:2009.11150*.

Ye Zhang, Iain Marshall, and Byron C Wallace. 2016. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 795–804.

Fei Zhao, Zhen Wu, and Xinyu Dai. 2020. Attention transfer network for aspect-level sentiment classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 811–821.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212.

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

A Appendix. Formula derivation.

The maximum entropy of the Gaussian distribution is equal to the result of the equation listed below.

$$
\begin{aligned}
&Maximize(H(\boldsymbol{\epsilon}_i)) \\
&= Maximize(-\int p(\boldsymbol{\epsilon}_i)\ln p(\boldsymbol{\epsilon}_i)d\boldsymbol{\epsilon}_i) \\
&= Maximize(\frac{1}{2}(\ln(2\pi{\sigma_i}^2)+1)) \\
&= Maximize(\ln 2(\frac{1}{2}\log(2\pi e)+\log\sigma_i)) \\
&= Maximize(\log\sigma_i)
\end{aligned}
$$

B Appendix. Details of experiment.

The statistics of widely-studied datasets used by different tasks are listed in Table 4. These datasets come from different topics, such as movie reviews, customer reviews, social reviews, and question type. In particular, since there is no standard partition of MR, CR, SUBJ, and MPQA, we follow the data splitting protocol, 7:1:2 for them to get the training, validation, and test sets. For the aspect-level tasks, we remove the instances with conflict sentiment labels in Laptop and Restaurant as implemented in (Peng et al., 2017).

The setup of hyperparameters for Att-BiLSTM and Memory Net are listed in Table 5. To make a fair compare with other algorithms, we set our hyperparameters the same as theirs.

C Appendix. Visualization and explanation.

In this section, we select several attention visualizations on SST2 test set to explain how PBSA works.

Pay more attention to important but low-frequency words Some words do have important effects on the results, but if they do not appear frequently enough then the traditional attention mechanism may not pay enough attention to them. As shown in Figure 5-(1), the word *drowsy* has an important influence on the emotional polarity of the film. However, it is a low-frequency word in the corpus, which makes the attention mechanisms do not allocate enough weights to it, resulting in a classification error. After being trained by PBSA, the model can assign enough weights to *drowsy*, which changes the result from false to correct.

Reduce the focus on high-frequency words that do not affect the results In baseline, some high-frequency words which do not contain any emotional polarity usually get high weights, while some important words that should have been focused on are ignored. As Figure 5-(2) shows, *romantic* and *doesn't* are words with strong emotional polarity. However, the baseline assigns greater weights to other high-frequency words (e.g., *between*) with no emotional polarity, and thus ignores the words *romantic* and *doesn't* which results in misclassification. After being trained by PBSA, the model reduces the focus on *between* and the weights allocated to the significant words increase correspondingly, which turns the result.

Increase the difference in weight between words with conflicting meanings As shown in Figure 5-(3), the baseline focuses on too many words: *horror*, *revenge*, *perfect*, *relentless*, *torture*, and so on. Maybe all of the words are important but the meanings of them are conflicting, which interferes with the classification task. The model feels confused because it does not know how to make a prediction according to so many emotional words. After being trained by PBSA, the difference in the weight of emotional words becomes larger, which makes it get the right result. It should be noted that the entropy of attention distribution may not decrease because PBSA keeps attention to important words while diluting the distribution of the other words.

Be more sensitive to adversative relations in sentences If there are adversative conjunctions (e.g., but, however, and so on) in the sentence, it is likely to express two opposite emotions before and after the adversative conjunction. This is when the model needs to keenly feel the changes of emotional polarity in the sentence. From this aspect, the model is also supposed to assign higher weights to those adversative conjunctions. Judging from our results, it is unfortunate that the original attention mechanism tends to ignore these conjunctions for they seem to have no effect on results outwardly. As Figure 5-(4) and Figure 5-(5) show, the baseline ignores the word *but* and results in errors. After being trained by PBSA, the baseline pays more attention to *but* which makes both of the emotions before and after the adversative conjunction can be taken into consideration.

Table 4: Detailed dataset statistics.

Task	Dataset	Class	AvgLen	Train	Test
Sentence Classification	SST2 (Socher et al., 2013)	2	19	6,920	1821
	TREC (Li and Roth, 2002)	6	10	5,452	500
	MR (Pang and Lee, 2005)	2	19	10,662	–
	CR (Hu and Liu, 2004)	2	19	3,775	–
	SUBJ (Pang and Lee, 2004)	2	23	10,000	–
	MPQA (Wiebe et al., 2005)	2	3	10,606	–
Document Categorization	IMDB (Maas et al., 2011)	2	280	25,000	25,000
Aspect-based Sentiment Analyis	REST (Pontiki et al., 2014)	3	16	3,591	1,121
	LAPTOP (Pontiki et al., 2014)	3	17	2,292	639
	TWITTER (Dong et al., 2014)	3	19	6,248	692

Table 5: Setup for Att-BiLSTM and Memory Net

Task	Dataset	Dimension of hidden states	Dimension of attention context
Sentence Classification	SST2 (Socher et al., 2013)	150	100
	TREC (Li and Roth, 2002)	150	50
	MR (Pang and Lee, 2005)	150	100
	CR (Hu and Liu, 2004)	150	50
	SUBJ (Pang and Lee, 2004)	150	100
	MPQA (Wiebe et al., 2005)	150	100
Document Categorization	IMDB (Maas et al., 2011)	150	300
Aspect-based Sentiment Analyis	REST (Pontiki et al., 2014)	300	300
	LAPTOP (Pontiki et al., 2014)	300	300
	TWITTER (Dong et al., 2014)	300	300

D Appendix. The illustration of the baselines.

D.1 Att-BiLSTM

Figure 6 shows the structure of Att-BiLSTM. Att-BiLSTM first map each word into pre-trained skip-gram (Mikolov et al., 2013) word embedding and then utilize 1-layered BiLSTM with a scale-dot attention mechanism to get sentence-level hidden states which are finally used for classification.

D.2 Att-BERT

Figure 7 shows the structure of Att-BERT. We add a scale-dot attention layer to the output of the BERT and use the output of the attention layer to classify.

D.3 Memory Network

Figure 8 shows the structure of MN. Memory Network uses an iteratively updated vector A (initialized as the aspect embedding) and the context embedding to generate the attention distribution, which is then used to select the important information from the context embedding and iteratively update the vector A.

D.4 BERTABSA

Figure 9 shows the structure of BERTABSA. We input the whole sentence to get the context representation of the aspect words, which is directly used for classification. To verify that our method truly improves the results, we delete the gating mechanism and use bert-base-uncased instead of bert-large-uncased.

D.5 Att-BERTABSA

Figure 10 shows the structure of Att-BERTABSA. Its structure is similar to Att-BERT, for adding a scale-dot attention layer after the output of BERT. However, different from Att-BERT, the hidden states of context words and aspect words are regarded as Q and K respectively and fed into the attention layer separately. To verify the effectiveness of our method, we make the same modifications on the Att-BERTABSA.

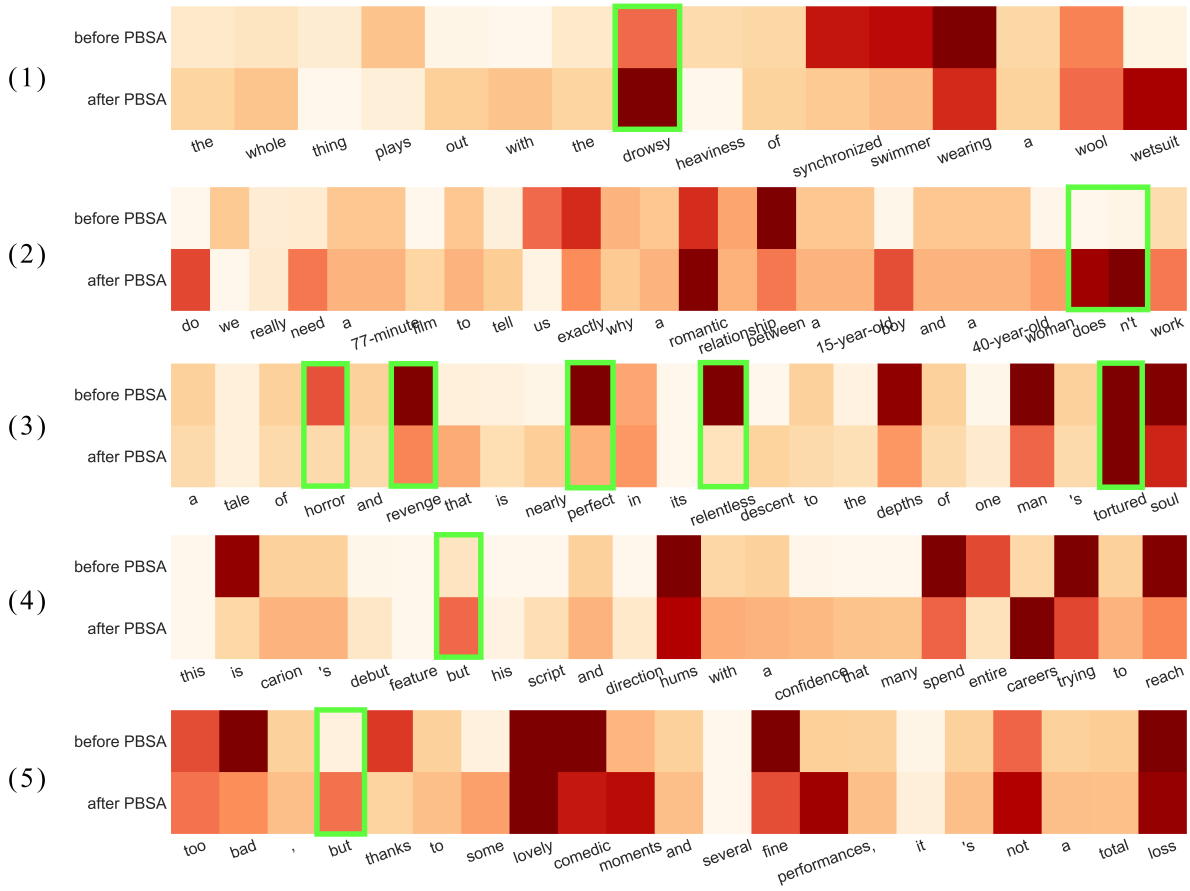

Figure 5: The visualization result of several samples on SST2 test set.

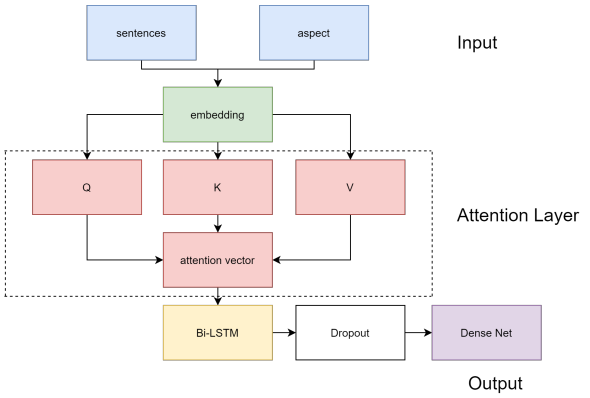

Figure 6: The illustration of Att-BiLSTM.

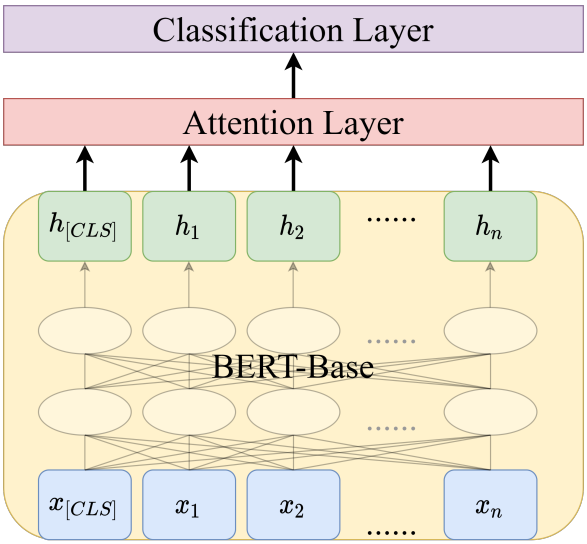

Figure 7: The illustration of Att-BERT.

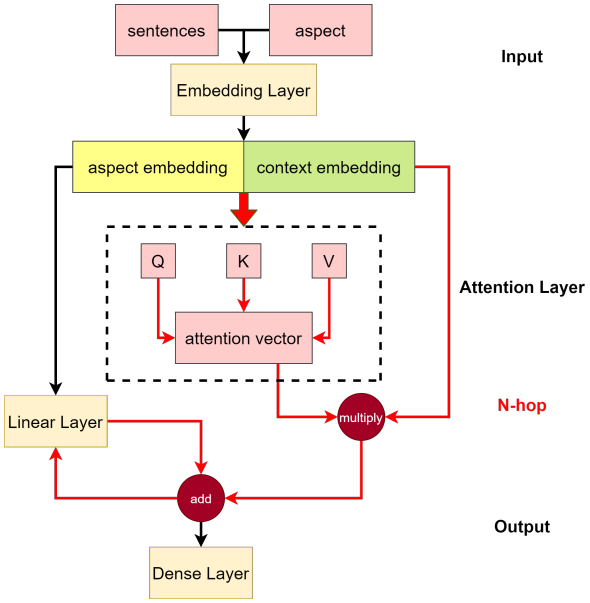

Figure 8: The illustration of Memory Net.

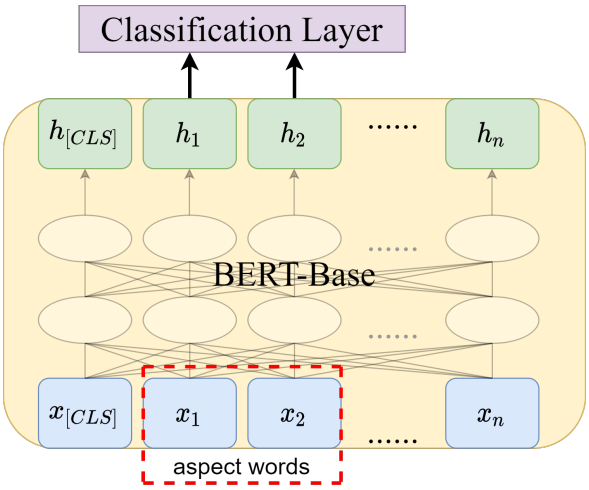

Figure 9: The illustration of BERTABSA.

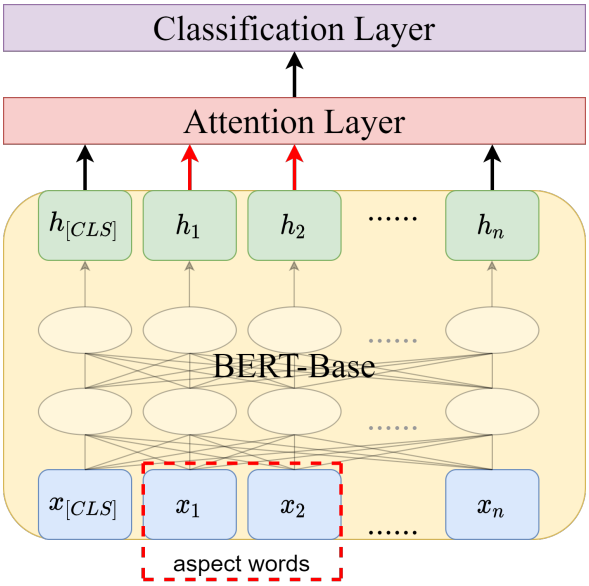

Figure 10: The illustration of Att-BERTABSA.